SENSATIONAL GAMES FOR THE AMSTRAD CPC464

JIM GREGORY

Introduction

Each game featured in these pages has been specially written for this book. This has resulted in a range of programs covering all types of games. They incorporate many useful techniques and some share the same subroutines. In some cases, control routines are arranged in such a way that if you use the method detailed in the chapter on subroutines you will be able to save yourself a lot of typing!

Each listing has been printed with a width setting of 40 columns to help reduce entry errors. This means that if you use Mode 1 to enter the programs then a line on the page will match a line on the screen; this feature is an example of the effort which has gone into ensuring your complete success with each and every program.

After extensive testing all the listings have been produced from actual working programs. Each has been reproduced directly from the computer printout. This means that the most likely reason for a program not working is that the reader has typed it in wrong!

I have entered many programs from books and magazines, and I know from experience that at some stage you will not believe that you have got it wrong. Please read the following notes on programming carefully. They will help you to avoid the most common mistakes as you enter the programs. Follow the advice given and you will find that your microcomputer hobby can be more enjoyable and less frustrating.

Although most of the following guidelines may seem obvious, each will save you time or trouble. If you are prepared and relaxed then few errors will be made.

First program yourself

1. Set up your equipment on a desk or table, make sure that you are seated properly and try to keep your back straight. Ensure that

lighting is good and that the book is supported where it can be read easily.

2. Before you start a long typing session ensure that everything is plugged in!

3. Make sure that you have a good supply of blank tapes.

4. If your blank tapes have a 'leader' on them, it is a good idea to prepare them for saving by turning the hub until the brown of the tape shows.

5. To make certain everything is OK, get a long audio tape such as a C90, which will be your working tape. Next complete a 'Save and load test', e.g. type in

```
10 REM  TEST
20 PRINT"TEST O.K."
30 GOTO 10
```

or a similar short program.

Save Test, rewind the tape, then load Test, and 'RUN' it. If this is OK, do not rewind the tape, leave it where it stopped. If all does not operate correctly then check everything and try again, until all is working. You will then be able to use this tape to store your program, either when you have finished or part way through to be continued later.

I recommend that you use the Speedwrite option and save the program twice rather than use the slow speed. This has been found to be totally satisfactory during the development of programs for this book.

6. When you save a part version give it a title which will help you later, such as 'V1', 'V2', etc. You may prefer to use the date and time of saving, e.g. 8.15-14/10/85.

As you save a version, remember to make a note on the cassette or insert of the title and the counter number. These will enable you to retrieve the program quickly when you wish to resume entering the program or to 'debug' your completed program.

7. Always save onto tape *before* you type RUN. The microworld is full of keen folk who lost hours of work because they typed RUN before saving. (If you have saved earlier versions as directed, at least only part will be lost.)

The reasons for losing a program in this way are various. Examples are: mistyping words so that the interpreter performs a 'new' instruction, or gets into a loop from which there is no escape except by switching off. Programs which incorporate either machine code,

POKE statements or System calls are most liable to 'crash', due to incorrect values being encountered by the program.

8. When a program is finished and working it is recommended that one copy is kept on a C90 back-up tape along with other programs. The main copy should be stored on a suitable short length cassette such as a C15. Only one program per side should be allowed, to enable it to be loaded easily and quickly when required. Remember to write the title on the label.

This two-tape system should help to prevent re-typing if a cassette is lost, damaged or develops a fault.

9. Many users become very careless about storage of cassettes and then wonder why they sometimes become faulty. If you want to keep what you've saved follow these rules:

- Identify all tapes and cases clearly.
- Keep them in cases in a rack or storage system.
- Store away from heat.
- Store away from any magnetic fields such as TV sets, computers, loudspeakers, motors, fans and very definitely telephones, all of which may damage or erase your program.
- Don't leave tapes in the deck when they are not required, and do not leave the deck switched into 'play' unless the program requires it.
- Remember to break out the plastic tabs at the back of the tape to prevent accidental erasure of the program. If ever you wish to re-record over a protected tape temporarily cover the tab hole with adhesive tape.

10. Don't spend longer than about three hours in a session. Have a break, walk about, have a shower, play a game!

Other problems

PARIS
IN THE
THE SPRING

Read the above out loud! You will be surprised to know that most people read it as 'Paris in the spring'. Perhaps you did too! However, it actually says 'Paris in the the spring'. The word 'the' is printed twice, but because it is not expected and because the phrase is familiar, the second 'the' is not 'seen' by the mind.

A similar problem occurs when programs are typed in from listings. You often type what you expect to see instead of what is printed. This is why the letter O is sometimes typed when it should be 'zero' (printed as Ø), or S is incorrectly typed instead of the dollar symbol (printed as $).

A moment's lapse of concentration can lead to lines becoming jumbled as the eye jumps from one line to the next. For example:

```
1Ø PRINT"This is the start"
2Ø PRINT"This is next"
```

becomes:

```
1Ø PRINT"This is the next"
```

Look at the screen to check that your shifted letters have been correctly entered. Failure to do this produces classics such as:

```
1Ø PRINT 2 WHERE ARE THE QUOTES!2
```

Under no circumstances should you attempt to alter a program as it is entered. If the numbers go: 1Ø,2Ø,3Ø, etc., then enter them like that. Changes are best made after a program is known to be working. This applies to everything – do not be tempted to miss out lines because you think that they may be optional. Otherwise program control could attempt to go to a line number that you have changed or deleted, and crash the program.

'Rem' statements are remark statements and have been used to help you to understand the program. They should be entered as printed. Only when a working program has been produced and saved should you perhaps produce a copy which has no 'REMs'. This could save a little memory, but there is no real benefit to be obtained. On the programs listed in this book it is safe to omit all 'REMs'. This is because care has been taken not to send control to a 'REM' line. Be careful with programs from other sources – there are some authors who think that it is good practice to send control to a 'REM' line.

Spaces are always of vital importance in a computer program. Care should be taken to get them right, to enable spaces to be counted and entered accurately. Simply align a piece of paper or card with the characters above and below, and count as the edge is moved along.

The width of printout has been set to match the 40 column Mode 1 display of the computer. If you use Mode 1 for entry this should help you to check that everything is right – for example, if the listing shows a line ending with four spaces. Then your line on the screen should also have four spaces at the end. Please be particularly careful to

enter each line as a complete line from the line number commands. Confusion could arise when the line includes numbers which wrap onto the next line. Do not enter these lines separately, or the computer will take it as a line number which will cause problems.

You may find it useful to place a ruler along the line you are up to. This will further reduce the risk of errors.

If you follow all the foregoing then you should not have any great problems. However, it is inevitable that some bugs will creep in. If your program fails to operate properly then the last chapter contains some helpful advice on debugging.

That's enough preparation for you, let's now start practising with the next chapter and some useful subroutines.

Techniques Used in the Programs

Colour pointers (logical colours)

The system of colour allocation used in the Amstrad can help produce the illusion of very fast movement. The system works by drawing each moving element of a picture separately with different colour pointers. Next the pointers are swapped under program control. When each colour is changed the effect on screen is instantaneous and produces the impression of movement.

Many of the games in this book have been designed around this feature. *Maze Maniac* uses the technique to great effect. The result is a maze which changes views very rapidly. This speed could not have been achieved otherwise, unless extensive machine code routines were used.

The cassette which came with your Amstrad has some other examples which are very decorative.

This technique works best in Mode 0, since 16 colours are available to produce in effect 16 sets of movement or switched zones. If greater resolution and limited movement is required then Mode 1 provides 4 sets.

Remember that one colour will be required for the background and any permanent information that you want on screen.

Take a careful look at the dice rolling routine contained in this book, to see another example of how effective this system can be.

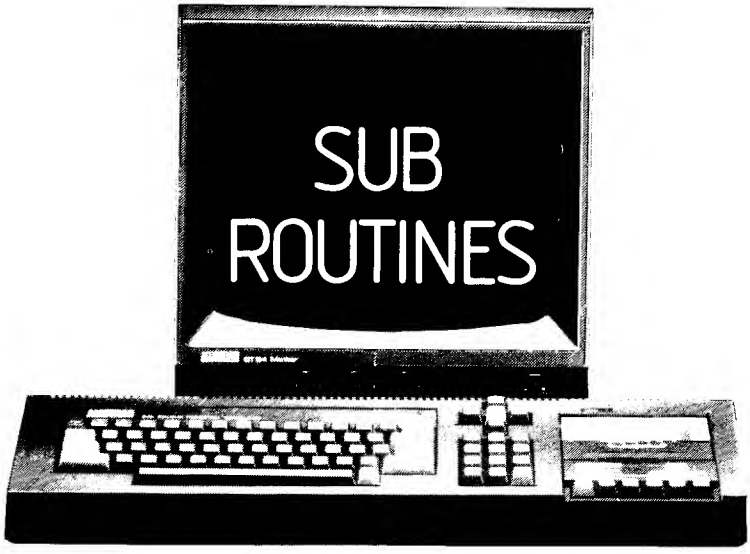
Exclusive oring (XOR)

This is another technique which is used extensively throughout this book. The use of XOR enables information to be superimposed upon the image which is already on screen. Performing the XOR again will remove the image, leaving the background as it was.

Examples of the commands required to implement this powerful feature can be found throughout the listings. The *Chess Duel* program contains some superb examples.

You will find it useful to use the listings as a source for solutions to programming problems. Try to understand the reason that certain routines are used and how they work. Once the programs are fully working and saved on tape then it is recommended that you experiment. Change the colours or parameters and see what happens. This way you can learn more about programming and be able to incorporate what you learn into your own programs. This is the most satisfying aspect of having your own computer.

Subroutines



Short Cuts to Success

We estimate that the total memory used by all the listings in this book is over 200K. It has therefore been essential to reduce the amount of entry to a minimum.

This has been done by the extensive use of *subroutines*. In particular three routines are used extensively.

1. High Score. This basic program takes the score achieved by our player and compares it with those already in its array. If the new score is higher than one in the table then the opportunity is given for the player to enter his nickname, initials or whatever. This routine should be typed in as listed here and then merged with the appropriate game when required.

Changes will be required to some programs, as follows. Type in the replacement lines given for each program.

Rainbow Breakout: 261ØØ GOSUB 6ØØØØ:GOTO 1ØØØ
Maze Maniac: 33Ø4Ø GOSUB 6ØØØØ:GOTO 1ØØØ
Parrot: 195 SC=PO*1Ø

2ØØ GOSUB 6ØØØØ:GOTO 25
Pick Man: 35Ø1Ø GOSUB 6ØØØØ:GOTO 1ØØØ
Kinkey Dong: 6ØØ5 GOSUB 6ØØØØ:GOTO 1ØØØ
Hi-Lo and *Skippy* need no additional lines. The subroutines should just be merged in.

```
'HIGH SCORE'


60000 MODE 1:BORDER 0:PAPER 0:PEN 2:CLS:
re=0
60060 PRINT:PRINT TAB(6);"High Scores fo
r ";gam$
60070 PRINT:PRINT
60080 FOR a=1 TO 5
60090 PRINT:PRINT TAB(4);a;".....";na$(a)
;TAB(30);sco(a)
60100 NEXT a
60110 FOR a=5 TO 1 STEP -1
60120 IF sc>sco(a) AND sc<sco(a-1) THEN
re=a
60130 NEXT
60140 IF re=0 THEN 60430
60150 FOR a=4 TO re STEP -1:na$(a+1)=na$
(a):sco(a+1)=sco(a):NEXT
60160 na$(re)="":sco(re)=sc
60170 CLS:PRINT:PRINT TAB(6);"High Score
s for ";gam$
60180 PRINT:PRINT
60190 FOR a=1 TO 5
60200 PRINT:PRINT TAB(4);a;".....";na$(a)
;TAB(30);sco(a)
60210 NEXT a
60220 PRINT:PRINT"        Please enter yo
ur name ":n$="ABCDEFGHIJKLMNOPQRSTUVWXYZ
 .]"
60230 PRINT:PRINT TAB(5);n$:x=5
60240 PRINT:PRINT"          ] ... End."
60250 LOCATE x,19:PRINT"-"
60260 LOCATE 10,22:PRINT na$(re)
60270 GOSUB 29000
60280 IF le=1 AND x>5 THEN LOCATE x,19:P
RINT " ":x=x-1
60290 IF ri=1 AND x<33 THEN LOCATE x,19:
PRINT " ":x=x+1
```

```
60300 IF fi=1 AND MID$(n$,(x-4),1)="]"TH
EN 60340
60310 IF fi=1 THEN na$(re)=na$(re)+MID$(
n$,(x-4),1)
60320 IF LEN(na$(re))=15 THEN 60340
60330 GOTO 60250
60340 MODE 1:CLS
60350 PRINT:PRINT TAB(6);"High Scores fo
r ";gam$
60360 PRINT:PRINT
60370 FOR a=1 TO 5
60380 PRINT:PRINT TAB(4);a;"....";na$(a)
;TAB(30);sco(a)
60390 NEXT a
60400 PRINT:PRINT "   Press FIRE to play
 again."
60405 a$=""
60410 GOSUB 29000:IF fi=1 THEN 60420
60415 GOTO 60410
60420 RETURN
60430 IF sc<sco(1) THEN 60340
60440 re=1:GOTO 60150
```

2. Inkeys. This is a machine code subroutine used to enable the joystick or the cursor pad to be read very quickly. The routine is poked into memory by a short program to avoid difficulties which may otherwise be encountered.

Type in the routine as listed and then save to tape. Then whenever required it can be merged into a program. Each listing clearly shows when the routine is required.

'INKEYS'

```
28980 REM >>>>>>>> inkeys/joy <<<<<<<<<
29000 le=0:ri=0:up=0:do=0:fi=0:ex=0:q=0
29010 CALL addr+1:a=PEEK(addr)
29020 IF a=8 OR a=242 THEN le=1
29030 IF a=9 OR a=243 THEN ri=1
29040 IF a=11 OR a=240 THEN up=1
29050 IF a=10 OR a=241 THEN do=1
29060 IF a=88 OR a=224 THEN fi=1
29070 IF a=13 THEN q=1
```

```
29080 IF a=32 THEN ex=1
29090 RETURN

29480 REM >>>>>>>> poke inkeys <<<<<<<<
29500 c=INT(addr/256):b=addr-256*c
29510 RESTORE 29580
29520 FOR n=addr TO addr+13
29530 READ a:IF a=999 THEN a=b
29540 IF a=998 THEN a=c
29550 POKE n,a
29560 NEXT n
29570 RETURN
29580 DATA 0,62,0,50,999,998,205,27
29590 DATA 187,208,50,999,998,201
```

3. Char Check. This is another machine code routine used extensively in this book. Its purpose is to place the ASCII value of a character cell into the BASIC variable 'CH'. It is used after the cursor has been positioned using LOCATE x,y. It is called by 'GOSUB 30000'.

This enables a program to detect whether two items have collided or perhaps whether the 'ball' in *Rainbow Breakout* has hit a brick. Once typed in and saved the routine is merged wherever indicated within a program listing.

```
'CHAR CHECK'


29980 REM >>>>>>>>> char check <<<<<<<<
30000 CALL loca+1
30010 ch=PEEK(loca)
30020 RETURN

30480 REM >>>>>>>>> poke check <<<<<<<<
30500 c=INT(loca/256):b=loca-256*c
30510 RESTORE 30580
30520 FOR n=loca TO loca+13
30530 READ a:IF a=999 THEN a=b
30540 IF a=998 THEN a=c
30550 POKE n,a
30560 NEXT n
30570 RETURN
30580 DATA 0,62,244,50,999,998,205,96
30590 DATA 187,208,50,999,998,201
```

Subroutines are used extensively for other programs in the book. These are listed along with the main programs where applicable.

Draughts/Chess. The board and other common routines can be merged as indicated.

Dice games. The dice throwing routine and the other three games are designed to form one whole program with menu selection for each game.

Card games. The 'card core' can be used in each game as directed.

How to merge

If it is wished to merge a set of routines into an existing program:

1. First save the program which is in the machine.
2. Next cue the tape containing the 'Merge' file.
3. Type MERGE "" and 'Enter'. Then press Play on the tape deck and follow prompts.
4. When complete, 'Save' the new program before 'Running'.
5. The program should now be ready.

Sometimes you may prefer to load in the subroutine program first and then simply add the new lines to it.

Once you have entered the subroutines and stored them on tape you are advised to try them out in a suitable program. A good one to start with is *Rainbow Breakout* in the classic games section.

1
Marie Celeste

An Adventure at Sea

In this text adventure you find yourself moored alongside the legendary ghost ship.

What is the secret of the *Marie Celeste*? Where are the crew?

By giving commands to the computer you can solve this marine mystery. By mixing up the text in the listing, care has been taken not to spoil your fun. In this way it is hoped that you will not be able to guess the solutions to the puzzles as they are typed in.

It is usual for adventures to require North, South, East and West as directions, but since this one has a nautical setting directions are: Forard, Aft, Starboard or Port. Which, in case you do not know, are equal to front of ship, rear of ship, right looking forward and left looking forward respectively.

As you move it is a good idea to draw a map. This map will help you to find your way around or identify locations that you have not yet visited.

Two-word instructions such as 'Take bucket' are accepted or the direction may be shortened to FOR, AFT, STA and POR. Adventures

often require the player to enter :'INVENTORY' to see what is being carried. In this adventure the inventory is permanently displayed. One of the enjoyable (but often frustrating) aspects of adventures is trying lots of different 'verbs' to see if they have any effect! Common verbs for adventures are: get, take, down, up, climb and unlock. Whilst you are on board, try lots of different ones. Try 'Examine cat' at the right time and see what happens.

Once you have solved the adventure invite others to play it. It is just as satisfying watching someone else make the same mistakes and perhaps have the same success as you.

```
1 REM <<< Marie Celeste <> ISSI/JIM >>>
2 REM
500 GOTO 29000
1000 RESTORE 1200
1010 FOR n=1 TO 6
1020 FOR m=2 TO 40
1030 READ a:LOCATE m,n
1040 PRINT CHR$(a);
1050 NEXT m
1060 NEXT n
1100 RETURN
1200 DATA 158,154,158,154,156,32,32,32,3
2,32,32,32,32,32,32,32,32,32,32,32,150,1
54,156,32,32,32,156,32,32,32,32,32,32,14
9,32,32,32,32,32
1210 DATA 149,32,149,32,149,32,32,32,32,
32,32,32,32,227,32,32,32,32,32,32,149,32
,149,32,32,32,149,32,32,32,32,32,32,151,
154,32,32,32,32
1220 DATA 149,32,149,32,149,32,150,154,1
56,32,158,154,156,158,32,150,154,156,32,
32,149,32,32,150,154,156,149,150,154,156
,150,154,156,149,32,150,154,156,32
1230 DATA 149,32,149,32,149,32,32,32,149
,32,149,32,32,149,32,151,154,153,32,32,1
49,32,32,151,154,153,149,151,154,153,149
,32,153,149,32,151,154,153,32
1240 DATA 149,32,149,32,149,32,150,154,1
57,32,149,32,32,149,32,149,32,32,32,32,1
49,32,32,149,32,32,149,149,32,32,147,154
,156,149,32,149,32,32,32
1250 DATA 155,32,155,32,155,32,147,154,1
55,32,155,32,32,155,32,147,154,153,32,32
```

```
,147,154,153,147,154,153,155,147,154,153
,147,154,153,147,153,147,154,153
1500 RETURN
19980 REM >>>>>>>>>> screen <<<<<<<<<<
20000 INK 0,3:INK 1,20:INK 2,0:INK 3,24
20010 MODE 1
20020 BORDER 0:PAPER 2:PEN 3:CLS
20030 WINDOW #1,1,30,8,16
20040 WINDOW #2,1,30,17,25
20050 WINDOW #3,31,40,8,25
20060 PAPER #1,0:PEN #1,1:CLS #1
20070 PAPER #2,2:PEN #2,3:CLS #2
20080 PAPER #3,3:PEN #3,0:CLS #3
20100 RETURN
20980 REM >>>>>>>>> initialize <<<<<<<<
21000 DIM d(10,7)
21010 DIM o$(11)
21020 DIM m(10,4)
21030 DIM o(11)
21100 RESTORE 21200
21110 FOR n=1 TO 10
21120 READ d(n,1),d(n,2),d(n,3),d(n,4)
21125 READ d(n,5),d(n,6),d(n,7)
21130 READ m(n,1),m(n,2),m(n,3),m(n,4)
21140 NEXT n
21150 FOR n=1 TO 11
21160 READ o$(n),o(n)
21170 NEXT n
21180 RETURN
21200 DATA 1,8,10,11,20,0,0
21205 DATA 0,0,0,0
21210 DATA 12,36,23,0,0,0,0
21215 DATA 1,1,0,0
21220 DATA 13,16,1,29,4,30,20
21225 DATA 1,0,1,0
21230 DATA 13,14,3,23,0,0,0
21235 DATA 0,1,1,0
21240 DATA 1,18,35,3,33,0,0
21245 DATA 0,1,0,0
21250 DATA 1,8,9,11,20,0,0
21255 DATA 0,0,0,0
21260 DATA 2,30,20,13,15,22,16
21265 DATA 1,0,1,1
21270 DATA 5,30,20,0,0,0,0
21275 DATA 0,0,1,0
```

```
21280 DATA 1,8,34,29,5,0,0
21285 DATA 1,0,0,1
21290 DATA 1,17,27,33,3,30,20
21295 DATA 0,0,1,0
21300 DATA KNIFE,6
21310 DATA TARPAULIN,1
21320 DATA WATER ON DECK,2
21330 DATA ROPE,6
21340 DATA NOTE,8
21350 DATA BUCKET,4
21360 DATA KEY,0
21370 DATA CAT,9
21380 DATA MOP,5
21390 DATA WELLIES,5
21400 DATA HOOKS,0
21980 REM >>>>>>>>>> input <<<<<<<<<<
22000 m$="":PRINT #2:PRINT #2,">";
22010 a$=INKEY$:a$=UPPER$(a$)
22020 IF (a$<"A" OR a$>"Z")AND a$<>" " A
ND a$<>CHR$(13) AND a$<>CHR$(127) THEN G
OTO 22010
22030 SOUND 1,150,2,15
22035 IF a$=CHR$(127) AND m$="" THEN GOT
O 22010
22040 IF a$=CHR$(127) THEN m$=LEFT$(m$,L
EN(m$)-1):PRINT #2,CHR$(8);" ";CHR$(8);:
GOTO 22010
22050 IF a$=CHR$(13) THEN PRINT #2,CHR$(
13):RETURN
22060 m$=m$+a$
22070 PRINT #2,a$;
22080 GOTO 22010
22980 REM >>>>>>>>> inventory <<<<<<<<<
23000 CLS #3
23010 PRINT #3,"INVENTORY.";
23020 PRINT #3,"----------"
23030 PRINT #3:PRINT #3
23040 FOR n=1 TO 11
23050 IF o(n)=13 THEN PRINT #3,o$(n)
23060 NEXT n
23070 RETURN
23980 REM >>>>>>>>>>> split <<<<<<<<<<<
24000 v$="":n$=""
24010 FOR n=1 TO LEN(m$)
24020 IF MID$(m$,n,1)=" " THEN GOTO 2406
```

```
0
24030 NEXT n
24040 v$=m$
24050 RETURN
24060 v$=LEFT$(m$,n-1)
24070 n$=RIGHT$(m$,LEN(m$)-n)
24080 RETURN
24980 REM >>>>>>>>>> command <<<<<<<<<<
25000 GOSUB 22000
25010 GOSUB 24000
25020 C$=LEFT$(v$,2)
25030 IF C$="PO" THEN GOSUB 30000
25040 IF C$="ST" THEN GOSUB 31000
25050 IF C$="FO" THEN GOSUB 32000
25060 IF C$="AF" THEN GOSUB 33000
25070 IF C$="GE" OR C$="TA" THEN GOSUB 3
4000
25080 IF C$="GO" OR C$="EN" THEN GOSUB 3
5000
25090 IF C$="EX" OR C$="IN" OR C$="RE" T
HEN GOSUB 36000
25100 IF C$="UN" OR C$="OP" THEN GOSUB 3
7000
25110 IF C$="CU" THEN GOSUB 38000
25120 IF C$="CL" THEN GOSUB 39000
25130 IF C$="KI" THEN GOTO 40000
25140 IF C$="LO" THEN GOSUB 40500
25150 IF C$="SH" OR C$="FU" OR C$="RA" T
HEN GOSUB 41000
25160 C$=LEFT$(V$,3)
25170 IF C$="SWA" OR C$="MOP" THEN GOSUB
 42000
25180 IF C$="MOV" OR C$="LIF" THEN GOSUB
 43000
25190 IF V$="QUIT" THEN GOTO 44000
25200 IF v$="`" THEN GOTO 25220
25210 PRINT #2,"I don't understand . . .
"
25220 GOSUB 28000:GOSUB 23000
25230 GOTO 25000
25980 REM >>>>>>>>>> start <<<<<<<<<<
26000 GOSUB 27000
26010 GOSUB 20000
26015 GOSUB 1000
26020 GOSUB 21000
```

```
26030 GOSUB 23000
26040 po=6
26100 RETURN
26980 REM >>>>>>>>>> strings <<<<<<<<<<
27000 DIM w$(36)
27010 RESTORE 27060
27020 FOR n=1 TO 36
27030 READ w$(n)
27040 NEXT n
27050 RETURN
27060 DATA I AM ,ASTERN,FORARD ,PORT ,ST
ARBOARD ,UP ,DOWN ,IN A ,SMALL ,LARGE
27070 DATA "BOAT, ALONGSIDE ",THERE IS ,
THE WIND ,HOWLS ,SINGS
27080 DATA SOUNDS LIKE LAUGHTER.,OUTSIDE
 ,INSIDE ,IT IS ,THE MARIE CELESTE.
27090 DATA A ,AND ,HERE.,AND I ,SLIDE ,A
CROSS ,LOCKED ,UNLOCKED ,TO ,OF ,IS ,I S
EE ,STORES.
27100 DATA GAP ,THE ,WATER ON THE DECK.,
DEVIL,BERMUDA TRIANGLE,ATLANTIS
27980 REM >>>>>>>>> describe <<<<<<<<<
28000 CLS #1
28005 IF po=2 AND o(10)<>13 THEN PRINT #
2,"You slip through to the":PRINT #2,"ne
xt deck section !":po=3:SOUND 2,300,10,1
5
28007 IF po=1 AND o(2)=0 THEN GOTO 45000
28010 FOR n=1 TO 7
28020 PRINT #1,w$(d(po,n));" ";
28030 NEXT n
28040 PRINT #1
28050 PRINT #1
28060 PRINT #1," I can see -"
28070 FOR n=1 TO 11
28080 IF o(n)=po THEN PRINT #1,o$(n);",
";
28090 NEXT n
28100 RETURN
28980 REM >>>>>>>> begin game <<<<<<<<<
29000 GOSUB 26000
29010 GOSUB 28000
29020 GOTO 25000
29980 REM >>>>>>>>> commands <<<<<<<<<<
```

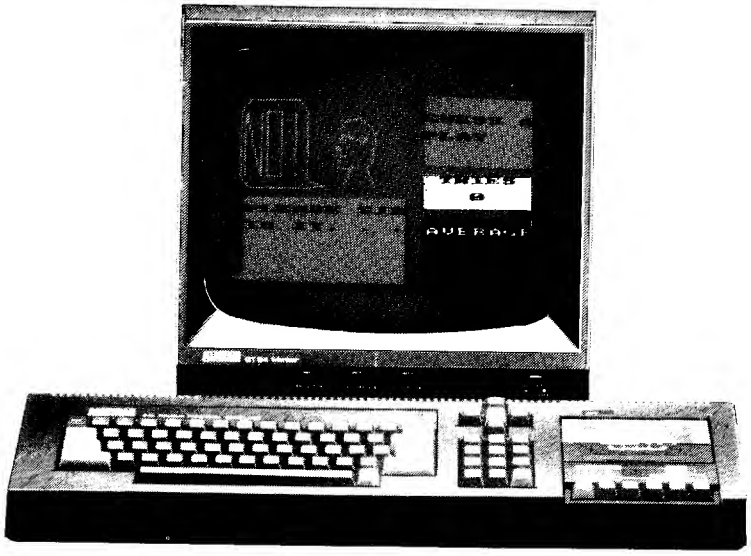
```
30000 v$="`"
30010 IF m(po,4)<>0 THEN GOTO 30020
30015 PRINT #2,"You cannot go that way."
:RETURN
30020 IF m(po,4)=1 THEN po=po-5:RETURN
30030 a=m(po,4):a=a-1:IF o(a)=13 THEN PR
INT #2,"You use the ";o$(a):PRINT #2,"an
d go to port." ELSE GOTO 30015
30040 FOR n=1 TO 3000:NEXT n
30050 po=po-5:RETURN
31000 v$="`"
31010 IF m(po,2)<>0 THEN GOTO 31020
31015 PRINT #2,"You cannot go that way."
:RETURN
31020 IF m(po,2)=1 THEN po=po+5:RETURN
31030 a=m(po,2):a=a-1:IF o(a)=13 THEN PR
INT #2,"You use the ";o$(a):PRINT #2,"an
d go to starboard." ELSE GOTO 31015
31040 FOR n=1 TO 3000:NEXT n
31050 po=po+5:RETURN
32000 v$="`"
32010 IF m(po,1)<>0 THEN GOTO 32020
32015 PRINT #2,"You cannot go that way."
:RETURN
32020 IF m(po,1)=1 THEN po=po+1:RETURN

32030 a=m(po,1):a=a-1:IF o(a)=13 THEN PR
INT #2,"You use the ";o$(a):PRINT #2,"an
d go to forard." ELSE GOTO 32015
32040 FOR n=1 TO 3000:NEXT n
32050 po=po+1:RETURN
33000 v$="`"
33010 IF m(po,3)<>0 THEN GOTO 33020
33015 PRINT #2,"You cannot go that way."
:RETURN
33020 IF m(po,3)=1 THEN po=po-1:RETURN
33030 a=m(po,3):a=a-1:IF o(a)=13 THEN PR
INT #2,"You use the ";o$(a):PRINT #2,"an
d go to aft." ELSE GOTO 33015
33040 FOR n=1 TO 3000:NEXT n
33050 po=po-1:RETURN
34000 v$="`"
34010 FOR n=1 TO 10
34020 IF n$=o$(n) THEN GOTO 34060
34030 NEXT n
```

```
34040 PRINT #2,"Don't be silly !"
34050 RETURN
34060 IF n=4 THEN PRINT #2,"It's tied to
o tightly for me !":RETURN
34070 o(n)=13
34080 PRINT #2,"You take the ";o$(n)
34090 RETURN
35000 v$="`"
35010 IF n$<>"STORE" AND N$<>"STORES" TH
EN PRINT #2,"You cannot go there . . .":
RETURN
35020 IF m(10,4)=0 THEN PRINT #2,"It's l
ocked !":RETURN
35030 po=5:RETURN
36000 v$="`"
36010 FOR n=1 TO 11
36020 IF n$=o$(n) THEN GOTO 36060
36030 NEXT n
36040 PRINT #2,"I don't see that here."
36050 RETURN
36060 IF n=1 THEN PRINT #2,"It's sharp !
"
36070 IF n=2 THEN PRINT #2,"It's green !
"
36080 IF n=3 THEN PRINT #2,"It's wet & s
alty !"
36090 IF n=4 THEN PRINT #2,"It's tied ti
ghtly."
36100 IF n=5 THEN PRINT #2,"It says -":P
RINT #2,"'No need to go below":PRINT #2,
"decks in this game.'"
36110 IF n=6 AND o(7)=0 THEN PRINT #2,"I
t's rattling !":o(7)=po
36120 IF n=6 AND o(7)<>0 THEN PRINT #2,"
It's empty !"
36130 IF n=7 THEN PRINT #2,"It says 'STO
RE'"
36140 IF n=8 THEN PRINT #2,"It runs off
with a dead rat.":o(8)=0
36150 IF n=9 THEN PRINT #2,"It's soggy !
"
36160 IF n=10 THEN PRINT #2,"The're smel
ly !"
36170 IF N=11 THEN PRINT #2,"They seem t
o be for a rope !"
```

```
36180 RETURN
37000 v$="`"
37010 IF po<>10 THEN PRINT #2,"I can't .
 .":RETURN
37020 IF o(7)<>13 THEN PRINT #2,"I have
no key !":RETURN
37030 m(po,4)=1:PRINT #2,"The store is o
pen."
37040 d(10,3)=28:RETURN
38000 v$="`":IF o(1)<>13 THEN PRINT #2,"
With what ?":RETURN
38010 IF n$=o$(4) THEN GOTO 38040
38020 PRINT #2,"You sadist !"
38030 RETURN
38040 PRINT #2,"You cut the rope and tak
e it."
38050 o(4)=13
38060 RETURN
39000 v$="`":IF n$<>o$(4) THEN v$="BI":R
ETURN
39010 IF o(4)<>po AND o(4)<>13 THEN PRIN
T #2,"Where's the ladder ?":RETURN
39020 PRINT #2,"You climb the rope ladde
r."
39030 IF po=6 THEN po=7:o(4)=7:RETURN
39040 IF po=7 THEN po=6:o(4)=6:RETURN
39050 IF po=1 THEN po=2:o(4)=2:RETURN

39060 RETURN
40000 PRINT #2," An Albatross flies down
"
40010 PRINT #2,"and attacks you for "
40020 PRINT #2,"being destructive."
40030 PRINT #2," You Are Dead ! "
40040 FOR n=1 TO 5000
40050 NEXT n
40060 SOUND 1,350,10,15
40070 GOTO 44000
40500 V$="`"
40510 PRINT #2:PRINT #2,"EXITS ARE ;"
40520 IF M(PO,1)<>0 THEN PRINT #2,"FORAR
D ";
40530 IF M(PO,3)<>0 THEN PRINT #2,"AFT "
;
40540 IF M(PO,2)<>0 THEN PRINT #2,"STARB
```

```
OARD ";
40550 IF M(PO,4)<>0 THEN PRINT #2,"PORT.
"
40560 IF m(po,1)=0 AND m(po,2)=0 AND m(p
o,3)=0 AND m(po,4)=0 THEN PRINT #2,"NONE
."
40570 PRINT #2
40580 RETURN
41000 v$="^"
41010 PRINT #2," Not now Sailor !"
41050 RETURN
42000 v$="^"
42010 IF o(6)<>13 OR o(9)<>13 THEN PRINT
 #2,"With what you fool . . .":RETURN
42020 IF po<>2 THEN PRINT #2,"Not here y
ou idiot . .":RETURN
42030 o(3)=0:PRINT #2,"Well done ! Now i
t is safe.":m(2,3)=5
42040 d(2,1)=1:d(2,2)=2:d(2,3)=20:d(2,4)
=32:d(2,5)=21:d(2,6)=11:d(2,7)=0
42050 o(11)=2:RETURN
43000 v$="^"
43010 FOR n=1 TO 10
43020 IF n$=o$(n) THEN GOTO 43060
43030 NEXT n
43040 PRINT #2,"Move what ? ! ? "
43050 RETURN
43060 PRINT #2,"You move it,and the"
43070 PRINT #2,"wind blows it away !"
43080 o(n)=0
43090 RETURN
44000 CLS #3:CLS #2
44010 PRINT #2,"Another Game ?"
44020 GOSUB 22000
44030 IF m$="YES" THEN RUN
44040 PRINT #2,"Well you're playing anyw
ay !"
44050 FOR n=1 TO 2500
44060 NEXT n
44070 RUN
45000 CLS #1
45010 PRINT #1,"SURPRISE !":PRINT #1,"We
 hid in ";w$(21);" ";w$(11);" ";w$(20)
45020 PRINT #1,"We thought You would"
45030 PRINT #1,"NEVER solve the adventur
e !"
45040 GOTO 44000
```

2
Please Sir



A Teaser for Everyone

This program is not quite what it seems! You may think that you are able to guess what the secret of the program is as you enter it, but you will still find the game a challenge when you play it.

The idea is that the on-screen 'teacher' selects a subject and you must then give examples which fit the subject. For example, if 'Films' was displayed then you might enter 'Please sir, is SUPERMAN one?' and teacher may say 'yes' or 'no'! You must try to decide why and then enter another example such as 'STAR WARS' to which the reply will once again be 'yes' or 'no'! The game continues until you achieve three 'yes' replies in a row, when it is assumed that you have realised why some get a 'yes' and others 'no'. You can cheat, but that most certainly would be a shallow victory.

Later, when you have mastered playing the program, you may wish to change the rules or add more subjects, and that may be a challenge in itself!

Invite a friend to list the program, and he may be surprised to find

that there is a distinct lack of data, whereupon you may either give your own reason or offer the following test of gullability:

> 'The program uses an advanced text compacting system based upon the mathematical characteristics of letters within English words. Thus, careful selection of parameters enables specific target answers to be 'tuned' into apparently imprecise test routines.'

Of course, you will know differently, because you typed in the lines which verify the entries! Why then, is a correct answer for the 'Birds' category: Sparrow, Chaffinch and Swallow?

```
1 REM <<<< please sir <> ISSI/JIM >>>>>
2 REM
500 SYMBOL 240,128,64,32,24,6,15,112,128

510 SYMBOL 241,1,3,7,31,127,255,15,1
1000 GOSUB 40500
1050 go=0
1100 GOSUB 20000
1150 GOSUB 22000
1160 GOTO 24000
19980 REM >>>>>>>>>> screen <<<<<<<<<<
20000 MODE 0
20010 PAPER 0:PEN 1:BORDER 6:CLS
20020 WINDOW #1,13,19,2,9
20030 WINDOW #2,13,19,11,14
20040 WINDOW #3,13,19,16,22
20050 WINDOW #4,2,11,14,23
20055 WINDOW #5,2,11,21,22
20060 RESTORE 20000
20070 FOR n=0 TO 15
20080 READ a:INK n,a
20090 NEXT n
20100 PAPER #1,3:PEN #1,4:CLS #1
20110 PAPER #2,2:PEN #2,3:CLS #2
20120 PAPER #3,8:PEN #3,1:CLS #3
20130 PAPER #4,9:PEN #4,4:CLS #4
20140 PAPER #5,9:PEN #5,4:CLS #5
20150 LOCATE 6,12:PRINT CHR$(240)
20160 PLOT 156,208:DRAW 40,208:DRAW 24,2
30
20170 DRAW 24,350:DRAW 40,372:DRAW 140,3
72
```

```
20180 DRAW 156,350:DRAW 156,225
20190 PLOT 35,350,13:DRAW 35,290:PLOT 65
,350:DRAW 65,230
20200 PLOT 107,350:DRAW 77,350:DRAW 77,2
30:DRAW 107,230
20210 PLOT 150,350,14:DRAW 120,350:DRAW
120,290:DRAW 150,290
20220 DRAW 150,230:DRAW 120,230:PLOT 39,
230:DRAW 61,230
20230 PLOT 39,290:DRAW 61,290:PLOT 77,29
0:DRAW 107,290
20240 PLOT 107,350,15:DRAW 107,230
20250 PLOT 135,348:DRAW 135,255:PLOT 135
,240:DRAW 135,232
20260 PLOT 35,232:DRAW 35,288:PLOT 39,34
8:DRAW 63,232
20270 PLOT 135,290,13
20280 PEN 9:LOCATE 1,23:PRINT CHR$(241)

20290 PEN 1
20300 LOCATE #1,1,3:PRINT #1,"GUESS A"
20310 LOCATE #2,2,1:PRINT #2,"TOTAL"
20320 LOCATE #3,2,2:PRINT #3,"GAMES"
20330 LOCATE #3,3,4:PRINT #3,go
20340 LOCATE #4,1,2:PRINT #4,"PLEASE SIR
":PRINT #4,"IS IT. . ."
20350 PLOT 290,210
20360 DRAW 275,250:DRAW 285,260
20370 DRAW 290,310:DRAW 285,320
20380 DRAW 260,340:DRAW 230,335
20390 DRAW 220,320:DRAW 220,310
20400 DRAW 230,305:DRAW 220,300
20410 DRAW 215,280:DRAW 220,282
20420 DRAW 216,275:DRAW 230,270
20430 DRAW 216,265:DRAW 220,260
20440 DRAW 216,256:DRAW 216,250
20450 DRAW 220,246:DRAW 260,248
20460 PLOT 240,246:DRAW 230,210
20470 TAG #7:MOVE 220,312
20474 PRINT #7,CHR$(240);
20475 MOVE 250,305
20478 PRINT #7,"?";:TAGOFF
20480 PLOT 225,308:DRAW 220,300
20485 GOSUB 21500
20490 RETURN
```

```
20500 DATA 6,24,2,18,1,0,15,16
20510 DATA 4,21,9,17,7,6,6,6
20980 REM
20990 REM * * * 'YES' * * *
21000 INK 13,24:INK 14,24:INK 15,6
21010 FOR n=500 TO 100 STEP -200
21020 SOUND 2,n,50,12
21030 NEXT n
21050 win=win+1
21200 RETURN
21480 REM
21490 REM * * * 'NO' * * *
21500 INK 13,24:INK 14,6:INK 15,24
21510 FOR n=200 TO 600 STEP 200
21520 SOUND 2,n,50,12,0,0,1
21530 NEXT n
21550 win=0
21700 RETURN
21980 REM >>>>>>>>> start game <<<<<<<<<
22000 win=0:go=go+1
22010 s=INT(RND(1)*20)+1
22020 LOCATE #1,1,5:PRINT #1,s$(s)
22030 LOCATE #2,3,3:PRINT #2,tr
22040 a=t(s,1):b=t(s,2):c=t(s,3)
22050 DEF FN ok=ASC(LEFT$(a$,1))
22060 IF a=2 THEN DEF FN ok=ASC(RIGHT$(a
$,1))
22070 IF a=3 THEN DEF FN ok=ASC(MID$(a$,
b,1))
22080 IF a=4 THEN DEF FN ok=LEN(a$)
22090 IF a=5 THEN DEF FN ok=sp+b
22100 IF s=2 THEN DEF FN ok=ASC(LEFT$(RI
GHT$(a$,2),1))
22250 RETURN
22980 REM >>>>>>>>>>>> input <<<<<<<<<<
23000 a$="":sp=0
23010 CLS #5:PRINT #5,a$
23020 w$=INKEY$:w$=UPPER$(w$)
23025 IF w$=" " THEN sp=sp+1:GOTO 23060
23030 IF w$=CHR$(13) THEN GOTO 23130
23040 IF w$=CHR$(127) AND LEN(a$)>0 THEN
 a$=LEFT$(a$,LEN(a$)-1):GOTO 23070
23050 IF w$<"A" OR w$>"Z" THEN GOTO 2302
0
23060 a$=a$+w$
```

```
23070 SOUND 2,200,10,12:GOTO 23010
23130 a=LEN(a$)
23140 IF a<1 THEN GOTO 23000
23150 SOUND 1,400,50,9:SOUND 1,200,20,9
23160 re=FN ok
23250 RETURN
23980 REM >>>>>>>>>>> loop <<<<<<<<<<<
24000 GOSUB 23000
24005 IF a$="QUIT" THEN RUN
24010 IF re=c THEN GOSUB 21000 ELSE GOSU
B 21500
24020 LOCATE #2,3,3:PRINT #2,win
24030 IF win=3 THEN GOTO 25000
24250 GOTO 24000
24980 REM >>>>>>>>>>> win <<<<<<<<<<<
25000 MODE 1
25010 LOCATE 12,3:PRINT "CONGRATULATIONS
 !"
25020 LOCATE 11,4:PRINT STRING$(18,45)
25030 LOCATE 9,10:PRINT "You solved the
puzzle !"
25100 LOCATE 8,23:PRINT "Press 'Y' to pl
ay again"
25110 a$=INKEY$:a$=UPPER$(a$)
25120 IF a$="Y" THEN GOTO 1100
25130 GOTO 25110
39980 REM >>>>>>>>>> subjects <<<<<<<<<
40000 DATA DOG,ACTOR,STAR,TV PROG,FILM,P
LAY,SONG
40010 DATA MICRO,COLOUR,FLOWER,TREE,GEM,
BOOK,SPORT,BEAST,BIRD,PLACE
40020 DATA FISH,NAME,CAR
40050 DATA 2,1,78,1,2,82,3,2,79,5,1,2
40060 DATA 1,1,83,5,0,0,5,5,6,4,1,8
40070 DATA 4,1,4,2,1,80,4,1,3,2,1,68
40080 DATA 5,2,4,3,3,67,3,4,83,3,3,65
40090 DATA 2,1,82,3,2,65,3,2,65,4,1,6
40500 RESTORE 40000
40510 DIM s$(20)
40520 FOR n=1 TO 20
40530 READ s$(n)
40540 NEXT n
40550 DIM t(20,3)
40560 FOR n=1 TO 20
40570 READ t(n,1),t(n,2),t(n,3)
40580 NEXT n
```

3
I Accuse

A Suspicious Listing

This is a test of deduction which will challenge all budding sleuths.

You are Inspector Watkins of CID (Computer Investigations Department) and are presently undertaking an extensive training course. This training is designed to test your courage and endurance. The course also tests your intelligence and ability to use the computer in criminal detection. Since we cannot cover the courage and endurance section in this book, we have included the computer deduction test.

The computer has the ability to generate thousands of different permutations from a set of data. In the blink of an eye it will decide on the criminal and his corroborative details, together with the statements of a witness that will conclusively point out the guilty party. You are allowed to ask for only six sets of details. This is done by entering the number corresponding to the category on the left of the screen. The computer will display them and then display the statements of key witnesses. This could be enough to identify the correct suspect, if you requested the right data!

The number of the suspect that you believe to be the criminal is entered and the accuracy of your judgement is reported back.

This program could well have educational value, but most of all it is designed to be different and fun.

```
1 REM <<<<<< I ACCUSE  ISSI/JIM >>>>>>>
2 REM
1000 GOSUB 22000
1200 GOSUB 20000
1500 GOSUB 21000
1600 FOR g=1 TO 6
1610 GOSUB 23000
1620 NEXT g
1630 CLS #2:PRINT #2,"You must now enter
 the number of the    suspect you consid
er to have commited   the crime."
1640 PRINT #2:PRINT #2,"   The witness s
aid that . . . . "
1650 RESTORE 1700
1660 FOR n=1 TO 10
1670 READ a$
1680 IF w(1)=n OR w(2)=n OR w(3)=n OR w(
4)=n THEN PRINT #2,"The criminal ";a$;"
";s$(s,n)
1690 NEXT n
1700 DATA was a,was,was,'s hair was,'s h
air was,wore a,'s shoes were,'s eyes wer
e,'s nose was,had a
1710 RESTORE 1720
1720 FOR n=1 TO 12
1730 READ a,b:SOUND 2,a,b,15
1740 NEXT n
1744 DATA 100,25,200,25,300,25,400,25
1746 DATA 200,25,300,25,400,25,500,25
1748 DATA 300,25,400,25,500,25,600,25
1750 a$=INKEY$:IF a$<"1" OR a$>"4" THEN
GOTO 1750
1760 cr=VAL(a$)
1770 SOUND 2,600,50,15:SOUND 2,400,50,15
:SOUND 2,200,50,15
1780 CLS #2
1790 IF cr=s THEN GOTO 3000
2000 PRINT #2,"You're pathetic, 'Inspect
or' Watkins .  No ! I will demote you to
 P.C. Watkins! "
```

```
2010 PRINT #2:PRINT #2
2020 PRINT #2,"It was suspect number ";s
;"!"
2050 GOTO 3050
3000 PRINT #2," Congratulations, Inspect
or Watkins !"
3010 PRINT #2:PRINT #2,"You correctly de
duced the true criminal from the clues g
iven."
3050 LOCATE 6,25:PRINT "Press any Key to
 Repeat Game"
3060 CALL &BB03
3070 a$=INKEY$:IF a$="" THEN GOTO 3070
3080 SOUND 2,100,50,15
3090 GOTO 1200
19980 REM >>>>>>>>>>screen<<<<<<<<<<<
20000 INK 0,0:INK 1,20:INK 2,6:INK 3,26
20010 BORDER 0:PAPER 0:PEN 1:MODE 1
20020 WINDOW #1,10,40,3,12
20030 WINDOW #2,1,40,15,24
20040 PAPER #1,0:PEN #1,1:CLS #1
20050 PAPER #2,0:PEN #2,3:CLS #2
20060 PEN 2:no=1
20070 FOR n=12 TO 36 STEP 8
20080 LOCATE n,1:PRINT no
20090 no=no+1
20100 NEXT n
20110 RESTORE 20100
20120 FOR n=3 TO 12
20130 LOCATE 1,n
20140 READ a$:PRINT a$
20150 NEXT n
20160 DATA 0-SEX,1-BUILD,2-HEIGHT,3-HAIR
,4-''COL,5-ATTIRE,6-SHOES,7-EYES,8-NOSE,
9-SPECIAL
20170 PRINT #2,"Choose your question (0-
9). . . ."
20500 RETURN
20980 REM >>>>>>>>>>set up<<<<<<<<<<<
21000 RANDOMIZE TIME
21010 s=INT(RND(1)*4)+1
21020 FOR n=1 TO 4
21022 w(n)=0
21024 check=0
21025 a=INT(RND(1)*10)+1
```

```
21030 FOR q=1 TO 4
21032 IF w(q)=a THEN check=1
21034 NEXT q
21035 IF check=1 THEN GOTO 21024
21037 w(n)=a
21040 NEXT n
21050 FOR n=1 TO 10
21060 FOR q=1 TO 4
21070 a=INT(RND(1)*4)+1
21080 s$(q,n)=l$(n,a)
21090 NEXT q
21100 NEXT n
21105 n=w(1):GOSUB 21150:n=w(2):GOSUB 21
150
21110 n=w(3):GOSUB 21150:n=w(4):GOSUB 21
250
21120 RETURN
21150 RANDOMIZE TIME:a=INT(RND(1)*4)+1
21160 s$(s,n)=l$(n,a)
21170 t=s+2:IF t>4 THEN t=t-4
21180 s$(t,n)=l$(n,a)
21190 IF a<>1 THEN b=a-1 ELSE b=4
21200 t=t-2:IF t<1 THEN t=t+4
21210 s$(t,n)=l$(n,b)
21220 t=t-2:IF t<1 THEN t=t+4
21230 s$(t,n)=l$(n,b)
21240 RETURN
21250 RANDOMIZE TIME:a=INT(RND(1)*4)+1
21260 s$(s,n)=l$(n,a)
21270 IF s<>1 THEN t=s-1 ELSE t=4
21280 s$(t,n)=l$(n,a)
21290 IF a<>4 THEN b=a+1 ELSE b=1
21300 t=t-1:IF t=0 THEN t=4
21310 s$(t,n)=l$(n,b)
21320 t=t-1:IF t=0 THEN t=4
21330 s$(t,n)=l$(n,b)
21340 RETURN
21500 DATA MAN,BOY,WOMAN,GIRL
21510 DATA FAT,THIN,SLIM,STUBBY
21520 DATA TALL,LARGE,SHORT,STOOPED
21530 DATA NORMAL,SHORT,CURLY,LONG
21540 DATA BROWN,BLACK,BLONDE,GINGER
21550 DATA JEANS,SUIT,JACKET,CLOAK
21560 DATA PUMPS,BOOTS,SANDLES,WELLIES
21570 DATA BLUE,BROWN,GREEN,HAZEL
```

```
21580 DATA SMALL,LARGE,BROKEN,CROOKED
21590 DATA TATOO,FRECKLE,DAGGER,GUN
21600 RETURN
21980 REM >>>>>>>>>initialize<<<<<<<<<
22000 DIM s$(4,10)
22010 DIM w(4)
22020 DIM l$(10,4)
22050 RANDOMIZE TIME
22060 RESTORE 21500
22070 FOR n=1 TO 10
22080 FOR m=1 TO 4
22090 READ l$(n,m)
22100 NEXT m
22110 NEXT n
22150 RETURN
22980 REM >>>>>>>>>>question<<<<<<<<<<
23000 a$=INKEY$:a$=UPPER$(a$)
23010 IF a$<"0" OR a$>"9" THEN GOTO 2300
0
23020 qw=VAL(a$)
23030 qw=qw+1
23035 SOUND 2,600-qw*50,25,15
23040 FOR pe=1 TO 4
23050 LOCATE #1,pe*8-7,qw
23060 PRINT #1,s$(pe,qw);
23070 NEXT pe
23200 RETURN
```

4
Star Trek

The Program that Goes Boldly On

There have been versions of *Star Trek* written for just about every computer ever made. This says a lot for its attraction as a game and makes this *the* classic game.

Now you can turn your Amstrad into a Federation Star Ship complete with scanners and phasers. Your mission is to clear the quadrant of the evil Klingons, without using up all your energy or being vaporised.

This version incorporates a zooming star field and 'real time' action during encounters with the enemy or when docking with star bases.

There follows an extract from the 32,544 page mega-manual entitled: 'How to be Captain of a Star Ship in 4000 easy lessons':

1. *Keep calm.* Remember the most you can lose is: your own life, 20 billion Galactic Dollars worth of hardware and the lives of everyone in the known universe.

2. *Watch where you are going!* Use the ship's eyes to reveal the nature of the universe. Pressing 'L' for Long Range Scan (LRS) will

display the position of your ship and the whereabouts of the enemy. Klingons are represented on the LRS by a circle with an arrow. This is the ancient symbol of Mars, the war bringer.

Star bases are represented by a circle above a cross. This is the ancient symbol for Venus, the bringer of love.

Those sectors which are occupied by *both* enemy and star bases have a square symbol in them. The square is an ancient symbol for a square.

3. *See what you can do.* Pressing 'S' will activate the short range scanners. This indicates the total number of enemy and star bases in the present sector.

4. *Keep an eye on your energy.* Docking with a star base will give you more. To dock press 'D' on your control panel. If there is a star base in that sector the docking sequence then requires you to steer your ship using left and right controls until the star base rests in the centre of your screen. Once docked you will receive 20 extra units to the overall ship's energy bank and a special 20 unit boost to your phasers and shields. Maximum capacity is 95 units. Failure to dock could result in the loss of the base. So be careful out there!

5. *Zap em', Cap'n!* A Star Fleet Captain has to do what he has to do, and that means eliminating the evil menacing Klingons.

If there are Klingons around, hitting the 'B' key sounds Battle Stations and you can start zapping. First get your enemy in the centre of the screen using the cursor pad or joystick. When you are ready, stab the copy key or fire button, sending phaser bolts out to destroy the target.

If you are not successful quickly you will lose the element of surprise and they will start to shoot back, thus draining your shield power. Do not give in, keep zapping until the last one is cleared from the sector. Then move on to win more battles.

6. *Warp carefully.* Movement to any sector is achieved by pressing 'W' followed by a number between 1 and 5. This is equal to the warp factor or distance you wish to travel.

Next the warp direction is required. This is a number between 1 and 8. Each number corresponds to a point on a compass, starting with one at the top and counting clockwise. Note: clocks have been retained in stardate 2806 to help Captains steer their Star Ships.

7. *The semi-final frontier.* On commencement of mission you can select the level. The easy level has only 10 enemies to seek and destroy. The average level has 20 and the hard level has 30.

As each level is completed, news of your success reaches Central Klingon Control and a replacement force, increased by 10 each time,

is sent out to do further battle. The ultimate challenge consists of 200 adversaries to wipe out. Are you ready Captain? Then let programming commence ...

```
1 REM <<<<<<<< star trek ISSI >>>>>>>>>
2 REM
500 SYMBOL 240,7,98,242,254,242,98,7,0
600 DIM s(15,2)
610 MEMORY 39995
620 addr=39999:GOSUB 29500
650 PRINT CHR$(23);CHR$(1);
1000 GOSUB 21000
1010 GOSUB 22000
1020 GOSUB 20000
1030 GOTO 4000
2000 a$=INKEY$:a$=UPPER$(a$)
2010 IF a$="L" THEN GOSUB 23000
2020 IF a$="S" THEN GOSUB 24000
2030 IF a$="W" THEN GOSUB 25000
2040 IF a$="D" THEN GOSUB 26000
2050 IF a$="B" THEN GOSUB 27000
2080 FOR n=1 TO 30:NEXT n
2090 GOSUB 11000
3000 IF a$="" THEN GOTO 2000
3010 IF dead=1 OR ener<1 THEN GOTO 7000
4000 LOCATE 7,21:PRINT ener
4010 LOCATE 7,23:PRINT shie
4020 LOCATE 7,25:PRINT phas
4030 LOCATE 17,21:PRINT sec
4040 LOCATE 17,23:PRINT enem
4050 LOCATE 17,25:PRINT s(sec,2)
4060 IF enem=alien THEN GOTO 10000
5000 GOTO 2000
6980 REM >>>>>>>>>>>> dead <<<<<<<<<<<<
7000 PAPER 1:PEN 3:CLS
7010 PRINT "    Bad Luck !"
7015 PRINT
7020 PRINT "        THE"
7025 PRINT
7030 PRINT "Starship Enterprise"
7035 PRINT
7040 PRINT " has been destroyed"
7045 PRINT
7050 PRINT "by the Klingons,but"
7055 PRINT
```

```
7060 PRINT " Admiral Sugar has"
7065 PRINT
7070 PRINT "decided to give you"
7075 PRINT
7080 PRINT " just 1 more chance"
7085 PRINT
7090 PRINT "if you want to get "
7095 PRINT
7100 PRINT " your vengeance !"
7110 LOCATE 3,25:PRINT "Any Key To Play"
7120 a$=INKEY$
7130 IF a$="" THEN GOTO 7120
7140 GOTO 1000
9980 REM >>>>>>>>>>> next <<<<<<<<<<<
10000 PAPER 1:PEN 3:CLS
10010 PRINT "    Well Done,"
10020 PRINT:PRINT "   Captain Kirk."
10030 PRINT:PRINT " Now you must fight"
10040 PRINT:PRINT " a fiercer group of"
10050 PRINT:PRINT " KLINGONS. . . . ."
10060 alien=alien+10
10070 LOCATE 3,25:PRINT "Any Key To Play
"
10080 a$=INKEY$
10090 IF a$="" THEN GOTO 10080
10100 GOTO 1010
10980 REM >>>>>>>>>> stars <<<<<<<<<<<
11000 INK clr,2
11010 clr=clr+1:IF clr=13 THEN clr=5
11020 INK clr,26
11030 RETURN
19980 REM >>>>>>>>>> screen <<<<<<<<<<
20000 MODE 0:SPEED INK 10,10
20020 INK 0,0:INK 1,2:INK 2,3:INK 3,24
20030 INK 4,3:INK 14,2,24:INK 15,24,6
20035 WINDOW #1,1,20,1,2
20040 WINDOW #2,2,13,3,19
20050 WINDOW #3,15,19,5,11
20060 WINDOW #4,15,19,13,19
20070 BORDER 0:PAPER 0:PEN 4:CLS
20080 PAPER #1,4:PEN #1,1:CLS #1
20090 PAPER #2,1:PEN #1,3:CLS #2
20100 PAPER #3,2:PEN #3,3:CLS #3
20110 PAPER #4,2:PEN #4,3:CLS #4
20120 LOCATE 16,4:PRINT "LRS"
```

```
20130 LOCATE 16,12:PRINT "SRS"
20140 PEN 3
20150 LOCATE 1,21:PRINT "ENERGY:"
20160 LOCATE 1,23:PRINT "SHIELD:"
20170 LOCATE 1,25:PRINT "PHASOR:"
20180 LOCATE 11,21:PRINT "SECTOR:"
20190 LOCATE 11,23:PRINT "ENEMY: "
20200 LOCATE 11,25:PRINT "*-BASE:"
20210 PEN 4
20290 PRINT CHR$(23);CHR$(0);
20300 col=5
20310 FOR n=1 TO 12
20320 PLOT 206+n*16,230,col
20330 PLOT 238-n*16,230,col
20340 PLOT 222,220+n*12,col
20350 PLOT 222,240-n*12,col
20360 PLOT 206+n*16,220+n*12,col
20370 PLOT 206+n*16,240-n*12,col
20380 PLOT 238-n*16,220+n*12,col
20390 PLOT 238-n*16,240-n*12,col
20410 INK 13,2
20420 PLOT 40,110,13:DRAW 200,220
20430 PLOT 400,110,13:DRAW 250,220
20440 col=col+1:IF col=13 THEN col=5

20450 NEXT n
20460 PEN #2,3
20470 LOCATE #2,5,12:PRINT #2,CHR$(147);
CHR$(158);CHR$(158);CHR$(153);
20490 clr=5
20500 PRINT CHR$(23);CHR$(1);
20510 RETURN
20980 REM >>>>>>>>> difficulty <<<<<<<<
21000 PAPER 0:PEN 1:MODE 1
21010 PRINT "Hard/Easy/Average ?"
21020 a$=INKEY$:a$=UPPER$(a$)
21030 IF a$="H" THEN alien=20:RETURN
21040 IF a$="E" THEN alien=10:RETURN
21050 IF a$="A" THEN alien=15:RETURN
21060 GOTO 21020
21980 REM >>>>>>>>> initialize <<<<<<<<
22000 RANDOMIZE TIME
22010 FOR n=1 TO 15:s(n,1)=0:s(n,2)=0:NE
XT n
22020 FOR n=1 TO alien
```

```
22030 sec=INT(RND(1)*15)+1
22040 IF s(sec,1)>3 THEN GOTO 22030
22050 s(sec,1)=s(sec,1)+1
22060 NEXT n
22070 FOR n=1 TO 3
22080 po=INT(RND(1)*15)+1
22090 IF s(po,2)=1 THEN GOTO 22080
22100 s(po,2)=1
22110 NEXT n
22120 phas=25:ener=25:bad=0:shie=25
22130 sec=INT(RND(1)*15)+1
22140 enem=0
22200 RETURN
22980 REM >>>>>>>>> long scan <<<<<<<<<
23000 SOUND 2,200,25,15:po=1
23010 FOR n=1 TO 3
23020 FOR m=1 TO 5
23030 LOCATE #3,m,n*2:PEN #3,3
23040 IF po=sec THEN PEN #3,15:PRINT #3,
CHR$(240):GOTO 23070
23045 IF s(po,1)>0 AND s(po,2)=1 THEN PE
N #3,4:PRINT #3,CHR$(232):GOTO 23070
23050 IF s(po,1)>0 THEN PRINT #3,CHR$(23
4):GOTO 23070
23055 PEN #3,14
23060 IF s(po,2)=1 THEN PRINT #3,CHR$(23
5)
23070 po=po+1
23080 NEXT m
23090 NEXT n
23100 FOR n=1 TO 7500:NEXT n
23140 CLS #3
23150 SOUND 2,200,25,15
23190 ener=ener-6
23200 RETURN
23980 REM >>>>>>>> short scan <<<<<<<<<
24000 SOUND 2,150,25,15
24005 PEN #4,15
24010 LOCATE #4,2,3:PRINT #4,CHR$(234);s
(sec,1)
24015 PEN #4,14
24020 LOCATE #4,2,5:PRINT #4,CHR$(235);s
(sec,2)
24030 FOR n=1 TO 7500:NEXT n
24040 CLS #4
```

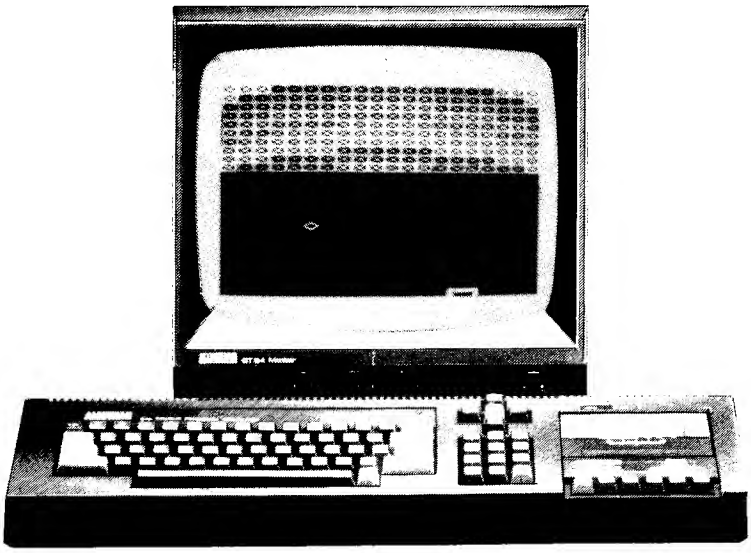
```
24050 SOUND 2,150,25,15
24090 ener=ener-3
24100 RETURN
24980 REM >>>>>>>>>>> warp <<<<<<<<<<<
25000 CLS #1:PEN #1,3
25010 PRINT #1,"WARP FACTOR (1-5) ?"
25020 a$=INKEY$:IF a$="" THEN GOTO 25020
25030 IF a$<"1" OR a$>"5" THEN SOUND 2,3
50,50,15:GOTO 25010
25040 dis=VAL(a$)
25050 CLS #1
25060 PRINT #1,"WARP DIRECTION ?"
25070 a$=INKEY$:IF a$="" THEN GOTO 25070
25080 IF a$<"1" OR a$>"8" THEN SOUND 2,3
50,50,15:GOTO 25060
25090 dir=VAL(a$)
25100 RESTORE 25500
25110 FOR n=1 TO dir
25120 READ xy
25130 NEXT n
25150 FOR n=1 TO dis
25160 sec=sec+xy
25170 IF sec<1 THEN sec=sec+15
25180 IF sec>15 THEN sec=sec-15
25190 SOUND 2,75*n,25,15
25250 NEXT n
25260 IF s(sec,1)>0 THEN GOSUB 25600
25270 CLS #1
25290 ener=ener-INT(dis*1.5)
25300 RETURN
25500 DATA -5,-4,1,6,5,4,-1,-6
25580 REM >>>>>>>>>>> alert <<<<<<<<<<<
25600 PEN #1,15
25610 a$="RED ALERT ! .RED ALERT ! .RED
ALERT ! . "
25620 x=INT(RND(1)*300)+100:dis=1
25650 FOR n=1 TO 25
25660 a$=RIGHT$(a$,39)+LEFT$(a$,1)
25670 LOCATE #1,1,1:PRINT #1,a$
25680 SOUND 2,100-n*2,1,15
25690 NEXT n
25700 RETURN
25980 REM >>>>>>>>>> docking <<<<<<<<<<
26000 CLS #1:PEN #1,3
26010 IF s(sec,2)=0 THEN PRINT #1,"No St
```

```
arbase . .":SOUND 2,300,25,15:RETURN
26020 x=INT(RND(1)*200)+75:dis=1
26030 y=350:TAG
26040 PLOT 0,0,1
26050 MOVE x,y:PRINT CHR$(235);
26060 GOSUB 28000
26070 MOVE x,y:PRINT CHR$(235);
26080 IF ri=1 THEN dis=dis-1
26090 IF le=1 THEN dis=dis+1
26100 y=y-1:x=x+dis
26102 IF x<30 THEN x=30:dis=dis+1
26104 IF x>380 THEN x=380:dis=dis-1
26110 IF y=200 THEN GOTO 26150
26120 GOTO 26050
26150 TAGOFF:PEN 4
26160 IF x<160 OR x>270 THEN PRINT #1,"M
ISSED ! !":SOUND 1,400,25,15:RETURN
26170 IF x>190 AND x<240 THEN GOTO 26200
26180 PRINT #1,"STARBASE DESTROYED":SOUN
D 1,500,50,15,0,0,31
26185 FOR n=5 TO 12:INK n,3:NEXT n
26190 FOR n=5 TO 12:INK n,6:NEXT n
26195 s(sec,2)=0:RETURN
26200 PRINT #1,"DOCKING . . . ."
26210 ener=ener+20:IF ener>95 THEN ener=
95
26215 phas=phas+20:IF phas>95 THEN phas=
95
26220 shie=shie+20:IF shie>95 THEN shie=
95
26230 SOUND 1,300,25,15:SOUND 1,200,25,1
5:SOUND 1,100,25,15
26240 FOR n=1 TO 2000:NEXT n:CLS #1
26250 RETURN
26980 REM >>>>>>>>>> battle <<<<<<<<<<
27000 FOR n=200 TO 50 STEP-1:SOUND 1,n,1
,15:NEXT n
27010 dead=0
27030 PEN #1,3
27040 IF s(sec,1)=0 THEN TAGOFF:CLS #1:P
RINT #1,"   SECTOR CLEAR.":RETURN
27050 x=INT(RND(1)*200)+75
27060 y=INT(RND(1)*100)+100
27070 TAG:PLOT 0,0,1
27100 TAG:MOVE x,y:PRINT CHR$(235);
```

```
27110 GOSUB 28000
27120 IF fi=1 AND phas>0 THEN GOTO 27500
27130 MOVE x,y:PRINT CHR$(235);
27132 TAGOFF
27135 IF kil=1 THEN s(sec,1)=s(sec,1)-1:
kil=0:GOTO 27010
27140 IF le=1 AND x<350 THEN xd=xd+1
27150 IF ri=1 AND x>45 THEN xd=xd-1
27160 IF do=1 AND y<350 THEN yd=yd+1
27170 IF up=1 AND y>110 THEN yd=yd-1
27180 IF ex=1 THEN RETURN
27200 x=x+xd:IF x<50 THEN x=50:xd=-xd
27210 IF x>360 THEN x=360:xd=-xd
27220 y=y+yd:IF y>360 THEN y=360
27230 IF y<120 THEN y=120:yd=-yd
27235 RANDOMIZE TIME
27240 IF RND(1)<0.95 THEN GOTO 27270
27250 INK 1,3:SOUND 1,400,20,15,0,0,31
27260 shie=shie-1:IF shie=-1 THEN dead=1
:RETURN
27265 FOR n=1 TO 30:NEXT n:INK 1,2
27270 LOCATE 7,23:PRINT shie;
27280 LOCATE 7,25:PRINT phas;
27290 LOCATE 17,23:PRINT enem;
27450 GOTO 27100
27490 REM * * * fire * * *
27500 INK 13,6
27510 FOR n=100 TO 50 STEP-1:SOUND 2,n,1
,15:NEXT n
27520 INK 13,2
27530 phas=phas-1
27540 IF y<240 AND y>200 AND x<240 AND x
>200 THEN enem=enem+1:kil=1:SOUND 2,300,
50,15,0,0,15
27600 GOTO 27130
27980 REM >>>>>>>>> key/star <<<<<<<<<<
28000 GOSUB 29000
28010 GOSUB 11000
28050 RETURN
```

Now 'MERGE' the 'INKEYS' routine. . .

5
Rainbow Breakout



An Arcade Classic

This second program classic has great attraction for players of all ages. It is one of the shorter programs in the book and so may be a good first attempt for those new to programming.

At the start you are asked to select a large or small bat, then the game begins. The object of the game is to eliminate all the blocks by hitting them with the ball. Your bat is moved left or right using the cursor pad or joystick. It must be moved in advance to anticipate the path of the ball as it bounces back from the blocks.

If you lose a ball then there are three supplied before the game restarts. The score is shown and can reach very high levels in the hands of a skilful player.

This game can be merged with the 'High Score' routine to provide a competitive game for more than one player or for a solo player to trace his hopefully improving play.

```
1 REM <<<<<<<<< rainbow ISSI >>>>>>>>>
2 REM
400 RANDOMIZE TIME
500 SYMBOL 240,255,195,165,153,153,165,1
95,255
510 SYMBOL 241,255,255,255,224,224,224,2
24,64
520 SYMBOL 242,255,255,255,0,0,0,0,0
530 SYMBOL 243,255,255,255,7,7,7,7,2
550 MEMORY 34995
560 addr=34999:GOSUB 29500
570 loca=35029:GOSUB 30500
1000 sc=0:li=5
1010 MODE 1
1020 INPUT "Small/Large ?",a$
1030 s$="":lim=19
1040 IF a$="l" OR a$="L" THEN  s$=CHR$(2
42)+CHR$(242):lim=17
1050 GOSUB 20000
1060 count=0
1100 x=1
1110 GOSUB 21100
1200 b=24
1210 RANDOMIZE TIME
1220 IF RND(1)>0.5 THEN c=1 ELSE c=0
1230 c=c+(INT(RND(1)*10)+5)
1250 bd=-1:ad=-1
1500 GOSUB 21000
1520 GOSUB 22000
1550 IF count>150 THEN GOTO 1050
2000 GOTO 1500
19980 REM >>>>>>>>>> screen <<<<<<<<<<
20000 RESTORE 20050
20010 FOR n=0 TO 15
20020 READ w
20030 INK n,w
20040 NEXT n
20050 DATA 0,24,18,26,20,4,6,15
20060 DATA 24,18,2,5,8,6,15,2
20070 PAPER 0:BORDER 2:MODE 0
20075 RESTORE 20210
20080 FOR n=1 TO 20
20090 READ col
20100 FOR m=1 TO 10
20110 LOCATE n,m
```

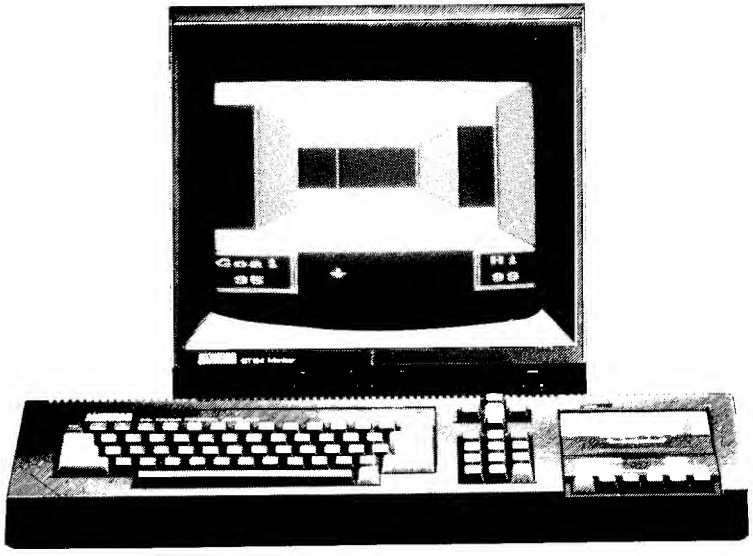
```
20120 PEN col
20130 PRINT CHR$(240);
20140 col=col+1
20150 NEXT m
20160 NEXT n
20200 RETURN
20210 DATA 3,4,4,5,5,5,5,6,6,6,6,6,6,5,5
,5,5,4,4,3
20980 REM >>>>>>>>> movement <<<<<<<<<
21000 GOSUB 29000
21010 IF le=1 AND x>1 THEN GOSUB 21400:x
=x-2:GOSUB 21100
21020 IF ri=1 AND x<lim THEN GOSUB 21400
:x=x+2:GOSUB 21100
21030 RETURN
21090 REM * * * draw * * *
21100 PEN 1
21110 LOCATE x,25:PRINT CHR$(241);s$;CHR
$(243);
21120 RETURN
21390 REM * * * rub * * *
21400 LOCATE x,25:PRINT "   ";
21420 IF s$<>"" THEN PRINT "   ";
21450 RETURN
21680 REM >>>>>>>>> draw ball <<<<<<<<<
21700 PEN 2
21710 LOCATE c,b:PRINT CHR$(202);
21720 RETURN
21790 REM >>>>>>>>>> rub ball <<<<<<<<<
21800 LOCATE c,b:PRINT " ";
21810 RETURN
21980 REM >>>>>>>> move ball <<<<<<<<<
22000 GOSUB 21800:a1=c:b1=b
22010 a1=a1+ad:b1=b1+bd
22020 IF a1=0 THEN a1=1:ad=1:SOUND 2,50,
1,15
22030 IF a1=21 THEN a1=20:ad=-1:SOUND 2,
50,1,15
22040 IF b1=0 THEN b1=1
22050 IF b1=25 THEN GOTO 22500
22060 LOCATE a1,b1:GOSUB 30000
22070 IF ch=240 THEN PRINT " ";:SOUND 2,
100,2,15:sc=sc+10:bd=-bd:count=count+1
22075 IF b1=1 THEN bd=1:SOUND 2,50,1,15
22080 c=c+ad:b=b+bd
```

```
22090 GOSUB 21700
22100 RETURN
22500 LOCATE c,25:GOSUB 30000
22510 IF ch=32 THEN GOTO 26000
22520 bd=-1
22530 IF RND(1)>0.5 THEN c=c+1
22540 SOUND 2,75,1,15
22550 RETURN
25980 REM >>>>>>>>>> dead <<<<<<<<<<
26000 GOSUB 21400:li=li-1
26010 SOUND 2,300,75,15
26020 IF li>0 THEN GOTO 1100
26100 MODE 0
26110 PEN 1
26120 PRINT " CONGRATULATIONS ! !"
26130 PRINT
26140 PRINT "You scored : "
26150 PRINT
26160 PRINT sc;" points !"
26500 LOCATE 1,24:PRINT "Another game ?
(Y/N)"
26510 a$=INKEY$:a$=UPPER$(a$)
26520 IF a$="Y" THEN GOTO 1000
26530 GOTO 26510
```

Now 'MERGE' the 'INKEYS' and the

'CHAR CHECK' routines. . .

6
Maze Maniac



A Game with Perspective

This third classic is a 3D maze game in which you have a perspective view of the paths open to you.

The object of the game is to find your way around the maze quickly. If you do it fast enough then you will catch the maniac. If you are too slow then *he* will catch *you*!

On screen are shown the passages that are open to you. The arrow at the bottom indicates the true direction that you are facing and how far you have travelled towards the exit at bottom right.

The left or right cursor changes the direction you are facing. The up cursor takes one step into the picture each time it is pressed. The down cursor takes you back one step and leaves you facing the same direction. The joystick can also be used to move.

Once you are successful in reaching the end of the maze then the target time is reduced and the best time must be beaten.

This game uses a sophisticated technique, based upon the Amstrad's colour pointers, to achieve the very fast changes to the picture. The technique is worthy of your close examination since you

can use it to produce high speed movement in your own programs.

If the program is merged with the 'High Score' routine then many players may race against each other's best time.

```
1 REM <<<< maze maniac <> ISSI/JIM >>>>
2 REM
500 SYMBOL AFTER 228
900 high=99
1000 GOSUB 24000
1020 GOSUB 22000
1050 po=1:view=4:targ=high-4
1060 LOCATE 9,22:PRINT "MAZE"
1065 LOCATE 8,24:PRINT "MANIAC"
1070 GOSUB 20000:GOSUB 32000
1080 GOSUB 27000:GOSUB 23000
1090 start=TIME
1100 GOSUB 28000
1150 GOSUB 27000
1160 GOSUB 23000
1170 IF won=1 THEN GOTO 33000
1180 won=0
1200 GOTO 1100
9980 REM >>>>>>>>>> monster <<<<<<<<<<
10000 SYMBOL 228,0,0,16,44,62,28,56,112
10010 SYMBOL 229,129,66,126,90,126,106,8
6,60
10020 SYMBOL 230,0,0,8,52,124,56,28,14
10030 SYMBOL 231,112,121,127,63,31,7,1,1
10040 SYMBOL 232,60,255,255,255,231,215,
235,213
10050 SYMBOL 233,14,158,254,252,248,224,
128,128
10060 SYMBOL 234,1,3,15,15,31,63,62,62
10070 SYMBOL 235,171,213,235,215,235,255
,60,0
10080 SYMBOL 236,128,192,240,240,248,252
,124,124
10090 SYMBOL 237,30,15,15,3,59,127,127,5
7
10100 SYMBOL 238,0,0,0,0,129,195,195,129
10110 SYMBOL 239,120,240,240,192,220,254
,254,156
10120 PEN 15:INK 15,13
10130 LOCATE 8,10:PRINT CHR$(228);CHR$(2
29);CHR$(230)
```

```
10140 LOCATE 8,11:PRINT CHR$(231);CHR$(2
32);CHR$(233)
10150 LOCATE 8,12:PRINT CHR$(234);CHR$(2
35);CHR$(236)
10160 LOCATE 8,13:PRINT CHR$(237);CHR$(2
38);CHR$(239)
10170 PEN 1
10200 RETURN
19980 REM >>>>> initialise screen <<<<<
20000 BORDER 0:PAPER 0:MODE 0
20040 INK 0,3:PEN 13
20050 PLOT 0,76,13:DRAW 639,76
20060 PLOT 0,72,13:DRAW 639,72
20070 PLOT 0,2,13:DRAW 639,2
20080 PLOT 0,6,13:DRAW 639,6
20090 PLOT 158,0:DRAW 158,78
20100 PLOT 150,0:DRAW 150,78
20110 PLOT 512,0:DRAW 512,78
20120 PLOT 520,0:DRAW 520,78
20130 GOSUB 21000
20140 PLOT 0,80,0:DRAW 240,200:DRAW 240,
278,15
20150 PLOT 639,80,0:DRAW 398,200:DRAW 39
8,278,15
20160 PLOT 160,160,14:DRAW 160,318
20170 PLOT 478,160,14:DRAW 478,318
20180 PLOT 80,120,13:DRAW 80,358
20190 PLOT 558,120,13:DRAW 558,358
20200 LOCATE 1,22:PRINT "Goal"
20210 LOCATE 1,24:PRINT targ
20215 PRINT CHR$(22);CHR$(1);
20220 LOCATE 18,22:PRINT "Hi"
20230 LOCATE 17,24:PRINT high
20240 PRINT CHR$(22);CHR$(0);
20270 RETURN
20980 REM >>>>>>>>> draw walls <<<<<<<<
21000 FOR n=80 TO 118 STEP 2
21010 PLOT 0,n,13:DRAW 638,n
21020 PLOT 0,280+n,13:DRAW 638,280+n
21030 NEXT n
21040 FOR n=120 TO 158 STEP 2
21050 PLOT 80,n,14:DRAW 558,n
21060 PLOT 80,200+n,14:DRAW 558,200+n
21070 NEXT n
21080 FOR n=160 TO 198 STEP 2
```

```
21090 PLOT 160,n,15:DRAW 478,n
21100 PLOT 160,120+n,15:DRAW 478,120+n
21110 NEXT n
21120 FOR n=200 TO 278 STEP 2
21130 PLOT 240,n,0:DRAW 398,n
21140 NEXT n
21150 FOR n=0 TO 78 STEP 4
21160 PLOT n,120,10:DRAW n,358
21170 PLOT 639-n,120,7:DRAW 639-n,358
21180 NEXT n
21190 FOR n=80 TO 158 STEP 4
21200 PLOT n,160,11:DRAW n,318
21210 PLOT 639-n,160,8:DRAW 639-n,318
21220 NEXT n
21230 FOR n=160 TO 238 STEP 4
21240 PLOT n,200,12:DRAW n,278
21250 PLOT 639-n,200,9:DRAW 639-n,278
21260 NEXT n
21270 FOR n=360 TO 398 STEP 2
21280 PLOT 78,360,1:DRAW 0,n
21290 PLOT 78,118,1:DRAW 0,n-280
21300 PLOT 560,360,4:DRAW 638,n
21310 PLOT 560,118,4:DRAW 638,n-280
21320 NEXT n
21330 FOR n=320 TO 358 STEP 2
21340 PLOT 158,320,2:DRAW 80,n
21350 PLOT 158,158,2:DRAW 80,n-200
21360 PLOT 480,320,5:DRAW 558,n
21370 PLOT 480,158,5:DRAW 558,n-200
21380 NEXT n
21390 FOR n=280 TO 318 STEP 2
21400 PLOT 238,280,3:DRAW 160,n
21410 PLOT 238,198,3:DRAW 160,n-120
21420 PLOT 400,280,6:DRAW 478,n
21430 PLOT 400,198,6:DRAW 478,n-120
21440 NEXT n
21450 RETURN
21980 REM >>>>>>>> colour data <<<<<<<<
22000 DIM l(14,9)
22010 RESTORE 22000
22020 FOR n=1 TO 14
22030 FOR m=1 TO 9
22040 READ a:l(n,m)=a
22050 NEXT m
22060 NEXT n
```

```
22070 RETURN
22080 DATA 6,6,6,6,6,6,20,11,2
22090 DATA 6,6,2,6,6,3,20,11,2
22100 DATA 6,11,6,6,3,6,20,11,2
22110 DATA 6,11,2,6,3,3,20,11,2
22120 DATA 20,6,6,3,6,6,20,11,2
22130 DATA 20,6,2,3,6,3,20,11,2
22140 DATA 20,11,6,6,3,6,20,11,2
22150 DATA 20,11,2,3,6,3,20,11,2
22160 DATA 6,6,3,6,6,3,20,11,3
22170 DATA 6,11,3,6,3,3,20,11,3
22180 DATA 20,6,3,3,6,3,20,11,3
22190 DATA 20,11,3,6,3,3,20,11,3
22200 DATA 6,3,3,6,3,3,20,3,3
22210 DATA 20,3,3,3,3,3,20,3,3
22980 REM >>>>>>> choose colours <<<<<<<
23000 FOR n=1 TO 3
23010 INK n,l(vl,n)
23020 NEXT n
23030 FOR n=4 TO 9
23040 INK n+6,l(vl,n)
23050 NEXT n
23060 FOR n=1 TO 6
23070 INK n+3,l(vr,n)
23080 NEXT n
23110 RETURN
23980 REM >>>>>>>>>> variables <<<<<<<<<
24000 DIM m(55)
24010 GOSUB 25000
24020 addr=34999:GOSUB 29500
24250 RETURN
24980 REM >>>>>>>>> maze data <<<<<<<<<
25000 RESTORE 25000
25010 FOR n=1 TO 55
25030 READ a:m(n)=a
25050 NEXT n
25090 RETURN
25100 DATA 4,6,1,6,7,5,4,6,5,6,1
25110 DATA 10,13,2,9,12,10,11,9,14,11,1
25120 DATA 6,9,6,3,9,2,5,6,9,6,5
25130 DATA 12,6,13,2,5,6,15,9,6,9,12
25140 DATA 10,9,10,3,9,8,10,3,9,2,9
25980 REM >>>>>>>>>> n/s/e/w <<<<<<<<<<
26000 x=m(po)
26010 no=0:so=0:ea=0:we=0
```

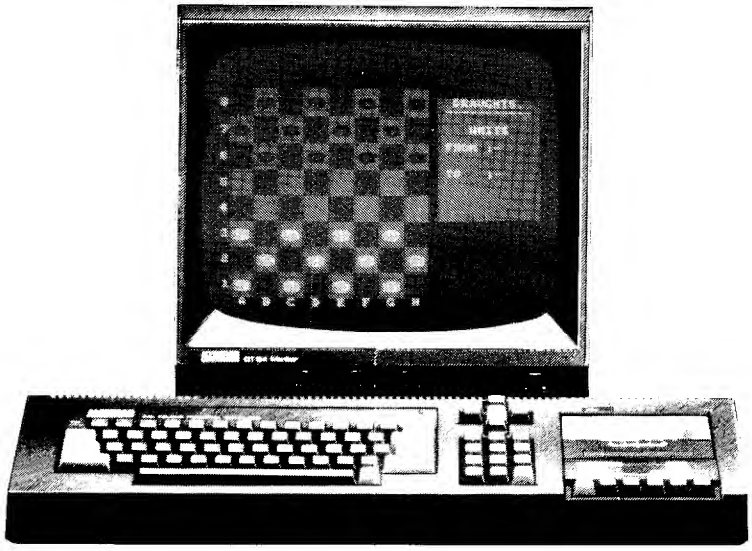
```
26020 IF x>=8 THEN no=1:x=x-8
26030 IF x>=4 THEN so=1:x=x-4
26040 IF x>=2 THEN ea=1:x=x-2
26050 IF x>=1 THEN we=1:x=x-1
26060 RETURN
26980 REM >>>>>>>>> check view <<<<<<<<<
27000 GOSUB 30000
27040 vl=1:vr=1:a=1:sa=po
27050 FOR q=2 TO 0 STEP -1
27055 ok=q*disp:IF sa+ok<1 OR sa+ok>55 T
HEN GOTO 27160
27060 po=sa+ok
27070 GOSUB 26000
27075 d=0:l=0:r=0
27080 IF view=1 THEN d=we:l=so:r=no
27090 IF view=2 THEN d=no:l=we:r=ea
27100 IF view=3 THEN d=ea:l=no:r=so
27110 IF view=4 THEN d=so:l=ea:r=we
27120 IF d=0 THEN GOSUB 27500
27130 IF l=1 THEN vl=vl+a
27140 IF r=1 THEN vr=vr+a
27150 a=a*2
27160 NEXT q
27170 po=sa
27180 RETURN
27480 REM >>>>>>>>>>>  XYZ <<<<<<<<<<<
27500 vl=13:a=1
27510 IF q=2 THEN vl=1
27520 IF q=1 THEN vl=9
27530 vr=vl
27540 RETURN
27980 REM >>>>>>>> move position <<<<<<
28000 GOSUB 29000
28010 IF le=1 THEN view=view-1:GOSUB 320
00
28020 IF ri=1 THEN view=view+1:GOSUB 320
00
28040 IF do=1 THEN view=view-1:GOSUB 320
00:view=view-1:GOSUB 32000:GOSUB 28500
28050 IF up=1 THEN GOSUB 28500
28060 IF do=1 THEN view=view-1:GOSUB 320
00:view=view-1:GOSUB 32000
28090 IF po=54 THEN WON=1
28110 RETURN
28500 IF view=1 AND we=0 THEN GOTO 28900
```

```
28510 IF view=2 AND no=0 THEN GOTO 28900

28520 IF view=3 AND ea=0 THEN GOTO 28900

28530 IF view=4 AND so=0 THEN GOTO 28900

28540 GOSUB 30000
28550 IF po+disp>55 OR po+disp<1 THEN GO
TO 28900
28560 po=po+disp:GOSUB 32000
28570 RETURN
28900 BORDER 18
28910 SOUND 2,300,25,15
28920 FOR z=1 TO 25
28930 NEXT z
28940 BORDER 0
28950 RETURN
29980 REM >>>>>>>>>> movement <<<<<<<<<
30000 IF view=1 THEN disp=-1
30010 IF view=2 THEN disp=-11
30020 IF view=3 THEN disp=1
30030 IF view=4 THEN disp=11
30040 RETURN
30980 REM >>>>>>>>>>> arrow <<<<<<<<<<<
31000 sym=241
31010 IF view=0 THEN view=4
31020 IF view=5 THEN view=1
31030 IF view=1 THEN sym=242
31040 IF view=2 THEN sym=240
31050 IF view=3 THEN sym=243
31070 PRINT CHR$(sym);
31080 RETURN
31980 REM >>>>>>>>>>>> map <<<<<<<<<<<<
32000 FOR n=21 TO 25
32010 LOCATE 6,n
32020 PRINT SPC(11):PRINT
32030 NEXT n
32040 yp=INT(po/11)
32050 xp=po-(11*yp)
32060 IF xp=0 THEN xp=11:yp=yp-1
32070 PRINT CHR$(23);CHR$(1);
32080 LOCATE xp+5,yp+21
32090 GOSUB 31000
32100 PRINT CHR$(23);CHR$(0);
32110 RETURN
```

```
32980 REM >>>>>>>>>>> out <<<<<<<<<<<
33000 fin=TIME-start
33010 sc=INT(fin/800)
33020 IF sc<=high THEN high=sc
33030 won=0:CLS
33040 PRINT:PRINT "Your Time Was ;";sc
33060 PRINT:IF sc<targ THEN PRINT "You g
ot your Goal !"
33070 GOSUB 10000
33490 LOCATE 3,24:PRINT "Now Try Again .
 ."
33500 FOR z=1 TO 3000:NEXT z
33510 GOTO 1050
```

Now 'MERGE' the 'INKEYS' routine. . .

7
Duo Draughts



Computer Moderated Micro Draughts

The idea is very simple - you must get rid of all your opponent's pieces, by jumping over them. To jump over a piece there has to be an empty space on the other side.

At the start you can only move in one direction, that is up the board if you are White or down if you are Black. To move a piece the letter for its column and the number for the row is entered. Next the destination square is given in the same way.

If it is not possible to jump over a piece then any one of your own draughts is advanced by one square to another square of the same colour. If a piece reaches the other side then it becomes a 'king' or 'crowner'. This means that it can move up and down the board and so is more powerful.

The fact that you have to take a piece when possible does not apply according to some sets of rules. Some authorities require that should an opponent's piece not be taken when possible, then the offender loses that piece. This is known as 'huffing'.

In this micro version the program will not accept an instruction

which simply moves a piece when there is a jump move available. The indicator will continue to show the same colour player until the jump sequence is complete.

A game can be saved to tape at any stage by pressing [shift] S. A previously saved game is loaded at any time by [shift] L. During a game or after loading, the step mode can be selected by pressing [shift] 3. Pressing 'Space' will then step through from the beginning. Pressing [shift] 3 will end step mode and allow the game to be played on from that position.

'BOARD CORE'

```
19980 REM >>>>>>>>>>> screen <<<<<<<<<<
20010 PAPER 0:PEN 1:BORDER 0:MODE 1
20020 WINDOW #1,2,25,1,24
20030 PAPER #1,3:PEN #1,2:CLS #1
20035 PAPER #2,2
20040 TAG #2
20070 l$=STRING$(3,143)
20080 l1$=STRING$(3,32)
20090 n$=l$+l1$+l$+l1$+l$+l1$+l$+l1$
20100 FOR n=1 TO 4
20110 FOR m=1 TO 3
20120 PRINT #1,n$;
20130 NEXT m
20140 FOR m=1 TO 3
20150 PRINT #1,RIGHT$(n$,21)+LEFT$(n$,3)
;
20160 NEXT m
20170 NEXT n
20200 FOR n=2 TO 23 STEP 3
20210 LOCATE 1,n
20220 l$=HEX$((26-n)/3,1)
20230 PRINT l$;
20240 LOCATE n+1,25
20250 PRINT CHR$(64+(n+1)/3);
20260 NEXT n
20270 WINDOW #3,27,39,2,15
20280 PAPER #3,3:PEN #3,1:CLS #3
23980 REM >>>>>>>>>>> input <<<<<<<<<<
24000 LOCATE #3,5,4:IF play=1 THEN PRINT
 #3,"WHITE" ELSE PRINT #3,"BLACK"
24010 LOCATE #3,10,6:PRINT #3,"  "
24020 LOCATE #3,10,9:PRINT #3,"  "
```

```
24030 LOCATE #3,10,6
24040 GOSUB 24500
24050 f$=n$
24060 LOCATE #3,10,9
24070 GOSUB 24500
24080 s$=n$
24100 RETURN
30980 REM >>>>>>>>>>>> save <<<<<<<<<<<
31000 z$(count,1)="FF"
31010 g$="":SOUND 2,200,20,15:SOUND 2,10
0,40,15
31020 FOR n=1 TO 8
31030 FOR m=1 TO 8
31050 NEXT m
31060 NEXT n
31065 h$="":i$=""
31070 FOR n=1 TO 100
31080 h$=h$+z$(n,1)
31085 i$=i$+z$(n,2)
31090 NEXT n
31100 h$=h$+STR$(count)
31120 PRINT #9,g$
31130 PRINT #9,h$
31140 PRINT #9,i$
31150 CLOSEOUT
31160 SOUND 2,200,20,15:SOUND 2,100,40,1
5
31200 RETURN
31980 REM >>>>>>>>>>>> load <<<<<<<<<<<
32000 g$="":SOUND 2,200,20,15:SOUND 2,20
0,40,15
32015 LINE INPUT #9,g$
32020 LINE INPUT #9,h$
32025 LINE INPUT #9,i$
32030 CLOSEIN
32035 FOR n=1 TO 8
32040 FOR m=1 TO 8
32070 NEXT m
32080 NEXT n
32090 FOR n=1 TO 100
32100 z$(n,1)=LEFT$(h$,2)
32110 h$=RIGHT$(h$,LEN(h$)-2)
32120 z$(n,2)=LEFT$(i$,2)
32125 i$=RIGHT$(i$,LEN(i$)-2)
32130 NEXT n
```

```
32140 count=VAL(h$)
32150 SOUND 2,200,20,15:SOUND 2,200,40,1
5
32160 GOSUB 20000
32170 GOSUB 23000
32180 LOCATE #3,10,6
32200 RETURN
```

'Board core' should now be saved separately to be used also in *Chess Duel*. The listing which follows completes the draughts 57 program.

```
1 REM <<<<<<< draughts <^> ISSI >>>>>>>
2 REM
50 SPEED WRITE 1
1000 GOSUB 21000
1010 FOR n=1 TO 100
1020 z$(n,1)="  "
1030 z$(n,2)="  "
1040 NEXT n
1100 GOSUB 20000:GOSUB 22000
1150 PRINT CHR$(23);CHR$(1);:GOSUB 23000
1160 count=1
1200 GOTO 1900
1600 gam=0
1610 FOR n=1 TO 8
1620 FOR m=1 TO 8
1630 IF m$(n,m)="B" OR m$(n,m)="D" THEN
gam=1
1640 NEXT m:NEXT n
1660 IF gam=0 THEN GOTO 10000
1670 gam=0
1680 FOR n=1 TO 8
1690 FOR m=1 TO 8
1700 IF m$(n,m)="A" OR m$(n,m)="C" THEN
gam=1
1710 NEXT m:NEXT n
1730 IF gam=0 THEN GOTO 11000
1750 RETURN
1900 GOSUB 29000
2000 GOTO 1900
9980 REM >>>>>>>>> black  win <<<<<<<<<
10000 CLS
10010 LOCATE 15,10:PRINT "BLACK won."
10050 GOTO 11050
10980 REM >>>>>>>> white  win <<<<<<<<<
```

```
11000 CLS
11010 LOCATE 15,10:PRINT "WHITE won."
11050 SOUND 2,300,25:SOUND 2,200,25:SOUN
D 2,100,25
11060 LOCATE 5,24:PRINT "Do You Want A R
eplay (Y/N) ?"
11070 a$=INKEY$:a$=UPPER$(a$)
11080 IF a$="Y" THEN GOSUB 35000:GOTO 11
00
11090 IF a$="N" THEN GOTO RUN
11100 GOTO 11060
20000 INK 0,0:INK 1,26:INK 2,14:INK 3,2
20290 PRINT #3,"  DRAUGHTS."
20300 PRINT #3," ------------"
20310 LOCATE #3,2,6:PRINT #3,"FROM :-"
20320 LOCATE #3,2,9:PRINT #3,"TO   :-"
20500 RETURN
20980 REM >>>>>>>> initialize <<<<<<<<<
21000 DIM m$(8,8)
21010 DIM z$(100,2)
21020 SYMBOL 240,0,0,0,0,15,127,255,255
21030 SYMBOL 241,0,0,0,0,240,254,255,207
21040 SYMBOL 242,188,175,170,170,122,15,
0,0
21050 SYMBOL 243,61,227,31,255,254,240,0
,0
21060 SYMBOL 244,0,1,9,87,99,118,125,255
21070 SYMBOL 245,0,128,144,234,198,110,5
8,187
21080 SYMBOL 246,191,175,170,170,122,15,
0,0
21090 SYMBOL 247,125,227,31,255,254,240,
0,0
21250 RETURN
21980 REM >>>>>>>> game start <<<<<<<<<
22000 FOR n=1 TO 8
22010 FOR m=1 TO 8
22020 m$(n,m)=" "
22030 NEXT m
22040 NEXT n
22050 FOR n=1 TO 8 STEP 2
22060 m$(1,n+1)="A"
22070 m$(2,n)="A"
22080 m$(3,n+1)="A"
22090 m$(8,n)="B"
```

```
22100 m$(7,n+1)="B"
22110 m$(6,n)="B"
22120 NEXT n
22190 RETURN
22980 REM >>>>>>>> draw pieces <<<<<<<<
23000 x=25:y=390
23010 FOR n=1 TO 8
23020 FOR m=1 TO 8
23030 c$=m$(n,m):c$=UPPER$(c$)
23040 IF c$=" " THEN GOTO 23100
23050 IF c$="A" THEN col=3 ELSE col=2
23070 PLOT 0,0,col
23080 MOVE x,y
23090 GOSUB 23500
23100 PLOT 0,0,0
23110 x=x+48
23130 NEXT m
23140 x=25:y=y-48
23150 NEXT n:RETURN
23490 REM * * * piece * * *
23500 d$=CHR$(240)+CHR$(241)
23510 d1$=CHR$(242)+CHR$(243)
23550 IF c$="C" OR c$="D" THEN d$=CHR$(2
44)+CHR$(245):d1$=CHR$(246)+CHR$(247):hu
ff=0
23600 PRINT #2,d$;
23610 MOVE x,y-16
23620 PRINT #2,d1$;
23750 RETURN
24500 n$=""
24510 a$=INKEY$
24520 IF a$="Q" THEN RUN
24524 IF a$="S" THEN GOSUB 31000
24526 IF a$="L" THEN GOSUB 32000
24528 IF a$="#" AND count>1 THEN z$(coun
t,1)="FF":GOSUB 35000
24530 a$=UPPER$(a$)
24540 IF a$<"A" OR a$>"H" THEN GOTO 2451
0
24550 n$=a$:SOUND 2,150,5,15:PRINT #3,n$
;
24560 a$=INKEY$
24570 IF a$<"1" OR a$>"9" THEN GOTO 2456
0
```

```
24580 n$=n$+a$:SOUND 2,150,5,15:PRINT #3
,a$;
24590 RETURN
24980 REM >>>>>>>>>>> move <<<<<<<<<<<
25000 GOSUB 26000
25090 ro=s1:co=s:GOSUB 25200
25100 m$(s1,s)=m$(f1,f)
25110 IF s1=1 AND m$(1,s)="B" THEN m$(1,
s)="D":huff=0
25120 IF s1=8 AND m$(8,s)="A" THEN m$(8,
s)="C":huff=0
25125 SOUND 2,200,75,15
25130 ro=s1:co=s:GOSUB 25200
25135 SOUND 2,100,10,15
25140 ro=f1:co=f:GOSUB 25200
25150 m$(f1,f)=" "
25160 RETURN
25200 x=25:y=390
25210 FOR n=1 TO 8
25220 FOR m=1 TO 8
25230 IF n=ro AND m=co THEN GOSUB 25300
25240 x=x+48
25250 NEXT m
25260 x=25:y=y-48
25270 NEXT n
25280 RETURN
25300 c$=m$(n,m)
25310 IF c$=" " THEN RETURN
25320 IF c$="A" OR c$="C" THEN col=3 ELS
E col=2
25340 PLOT 0,0,col:MOVE x,y
25350 GOSUB 23500
25360 RETURN
25980 REM >>>>>>>>>> convert <<<<<<<<<
26000 f=ASC(LEFT$(f$,1))
26010 f=f-64
26020 f1=VAL(RIGHT$(f$,1))
26030 f1=9-f1
26040 s=ASC(LEFT$(s$,1))
26050 s=s-64
26060 s1=VAL(RIGHT$(s$,1))
26070 s1=9-s1
26080 RETURN
26090 REM >>>>>>> convert back <<<<<<<<
26100 r$=CHR$(col+64)
```

```
26110 r$=r$+STR$(9-lin)
26120 RETURN
26980 REM >>>>>>>>>> validate <<<<<<<<<
27000 GOSUB 26000:huff=0
27005 p$="B":p1$="D"
27007 IF play=2 THEN p$="A":p1$="C"
27010 FOR y=1 TO 8
27020 FOR x=1 TO 8
27030 IF huff=1 THEN GOTO 27060
27040 IF m$(y,x)<>p$ AND m$(y,x)<>p1$ TH
EN GOTO 27060
27050 GOSUB 33000
27060 NEXT x
27070 NEXT y
27075 check=1
27080 c$=m$(f1,f):d$=m$(s1,s)
27090 IF c$=" " THEN RETURN
27100 IF (c$="A" OR c$="C") AND play=1 T
HEN RETURN
27110 IF (c$="B" OR c$="D") AND play=2 T
HEN RETURN
27120 IF d$<>" " THEN RETURN
27130 IF c$="A" AND s1<f1 THEN RETURN
27140 IF c$="B" AND s1>f1 THEN RETURN
27150 xd=s-f:yd=s1-f1
27160 IF SGN(xd)*(xd)=1 AND SGN(yd)*(yd)
=1 THEN GOTO 27490
27170 IF SGN(xd)*xd<>SGN(yd)*yd THEN RET
URN
27180 x$=m$(f1+(yd/2),f+(xd/2))
27190 IF (x$="A" OR x$="C") AND play=2 T
HEN RETURN
27200 IF (x$="B" OR x$="D") AND play=1 T
HEN RETURN
27210 IF x$=" " THEN RETURN
27220 ro=f1+(yd/2):co=f+(xd/2):GOSUB 252
00
27230 lin=f1+(yd/2):col=f+(xd/2)
27240 m$(lin,col)=" "
27250 GOSUB 26100
27260 z$(count,1)=r$:z$(count,2)=r$
27270 count=count+1:GOTO 27500
27490 IF huff=1 THEN RETURN
27500 check=0:RETURN
27980 REM >>>>>>>>>>> turn <<<<<<<<<<<
```

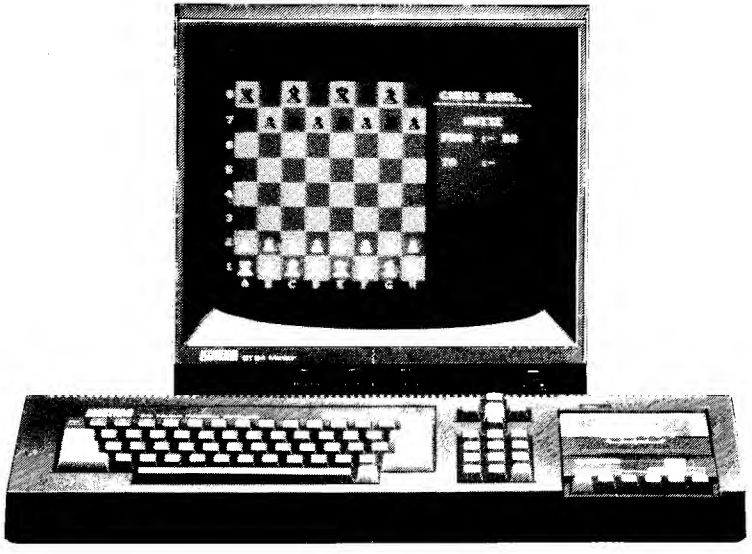
```
28000 GOSUB 24000:huff=0
28005 LOCATE #3,1,11:PRINT #3,SPC(12)
28010 GOSUB 27000
28020 IF check=0 THEN GOTO 28050
28030 LOCATE #3,1,11:PRINT #3,"INVALID M
OVE"
28040 SOUND 2,300,50,15:GOTO 28000
28050 GOSUB 25000
28060 z$(count,1)=f$:z$(count,2)=s$
28070 count=count+1:IF count=101 THEN co
unt=1
28075 IF huff=0 THEN GOTO 28100
28080 y=s1:x=s:GOSUB 33000
28090 IF huff=1 THEN GOTO 34000
28100 huff=0:RETURN
28980 REM >>>>>>> both players <<<<<<<<
29000 play=1
29010 GOSUB 28000
29020 GOSUB 1600
29030 play=2
29040 GOSUB 28000
29050 GOSUB 1600
29060 RETURN
31040 g$=g$+m$(n,m)
31110 OPENOUT "!DRAUGHTS"
32010 OPENIN "!DRAUGHTS"
32050 m$(n,m)=LEFT$(g$,1)
32060 g$=RIGHT$(g$,LEN(g$)-1)
32980 REM >>>>>>>>>>> huff <<<<<<<<<<<<
33000 huff=0
33010 h1$="A":h2$="C"
33020 IF play=2 THEN h1$="B":h2$="D"
33030 IF y<3 OR m$(y,x)="A" THEN GOTO 33
080
33040 IF x<3 THEN GOTO 33060
33050 IF (m$(y-1,x-1)=h1$ OR m$(y-1,x-1)
=h2$) AND m$(y-2,x-2)=" " THEN huff=1
33060 IF x>6 THEN GOTO 33080
33070 IF (m$(y-1,x+1)=h1$ OR m$(y-1,x+1)
=h2$) AND m$(y-2,x+2)=" "  THEN huff=1
33080 IF y>6 OR m$(y,x)="B" THEN GOTO 33
130
33090 IF x<3 THEN GOTO 33110
33100 IF (m$(y+1,x-1)=h1$ OR m$(y+1,x-1)
=h2$) AND m$(y+2,x-2)=" " THEN huff=1
```

```
33110 IF x>6 THEN GOTO 33130
33120 IF (m$(y+1,x+1)=h1$ OR m$(y+1,x+1)
=h2$) AND m$(y+2,x+2)=" " THEN huff=1
33130 RETURN
33980 REM >>>>>>>>> go again <<<<<<<<<
34000 LOCATE #3,10,6:PRINT #3,s$
34010 f$=s$
34020 LOCATE #3,10,9:PRINT #3,"  "
34030 LOCATE #3,10,9
34040 GOSUB 24500
34050 s$=n$
34060 LOCATE #3,1,11:PRINT #3,SPC(12)
34070 GOSUB 26000:GOSUB 27075
34080 IF check=0 THEN GOTO 34110
34090 LOCATE #3,1,11:PRINT #3,"INVALID M
OVE"
34100 SOUND 2,300,50,15:GOTO 34020
34110 GOSUB 25000
34120 z$(count,1)=f$:z$(count,2)=s$
34130 count=count+1:IF count=101 THEN co
unt=1
34140 y=s1:x=s:GOSUB 33000
34150 IF huff=1 THEN GOTO 34000
34160 huff=0:GOTO 28150
34980 REM >>>>>>>>>> replay <<<<<<<<<<
35000 count=0:play=1
35010 GOSUB 22000
35020 GOSUB 20000:GOSUB 23000
35030 count=count+1
35040 IF z$(count,1)="FF" THEN GOTO 3550
0
35050 f$=z$(count,1):s$=z$(count,2)
35060 GOSUB 26000:GOSUB 25000
35065 IF f$=s$ THEN GOTO 35030
35070 count=count+1
35080 a$=INKEY$
35090 IF a$="" THEN GOTO 35080
35095 play=2
35100 IF a$="#" THEN GOTO 35500
35110 IF z$(count,1)="FF" THEN GOTO 3550
0
35120 f$=z$(count,1):s$=z$(count,2)
35130 GOSUB 26000:GOSUB 25000
35135 IF f$=s$ THEN GOTO 35030
35140 a$=INKEY$
```

```
35150 IF a$="" THEN GOTO 35140
35155 play=1
35160 IF a$="#" THEN GOTO 35500
35170 GOTO 35030
35500 IF play=1 THEN GOTO 29000
35510 GOTO 29030
```

Now 'MERGE' the 'BOARD CORE' routine. .

8
Chess Duel



Computer Moderated Micro Chess

It will come as a surprise to many that before the age of chess computers and the microcomputer program, the game of chess was originally played by two human beings.

An intelligent chess playing program would take up an entire book on it's own, so it was decided to develop a program which would be of genuine use to anyone who enjoyed playing chess or would like to learn more about it. Chess clubs could also find the program helpful.

The program can be used simply as a replacement for a chess board and pieces. The two players enter their moves in usual standard chess notation. However, extra features have been included to produce worthwhile advantages over the traditional board and pieces.

The computer will examine all entries for legality of the move. Only moves which are the correct colour, are from occupied squares and move to a square which can be reached by that piece, will be accepted. Pieces which are 'taken' in the process are removed from the board.

The program does not indicate 'check' or 'mate' and will allow

moves which reveal check. It is up to the humans to watch out for these. *En passant* is not automatically dealt with but the 'remove' option may be used by a player to take the pawn if required on his move.

Castling is handled by the program. The present position of the king and rook have to be correct, with no pieces between them, for castling to be implemented. The routine will still castle if a piece has previously been moved but returned to the correct square. Moving through check is not monitored. These latter conditions must be detected by the players.

A useful feature is that the initial board may be set up for chess problems or chess variants and games of up to 100 moves can be stored and analysed step by step. Play can continue from any stage of a saved game or the present game can be re-started and then played from any subsequent move. Pawns are automatically promoted to a queen when they reach the appropriate rank.

Summary of commands and options

1. Set up board. Enter 'Y' when asked. Then sequentially from the top left, place each piece using a single letter. Move on using 'Space'.

   White pieces: P=pawn, R=rook, N=knight, K=king, Q=queen, B=bishop.
   Black pieces: the same but with 'Shift' held.

White always plays up the board.

2. Remove piece. 'Shift' 'R' then enter the position. Play continues with the next player.
3. Castle. 'Shift' 'C' then direction 'L' or 'R'.
4. Save game. 'Shift' 'S'. You must ensure cassette Record and Play are on before saving as prompts are not displayed.
5. Load Game. 'Shift' 'L', then press 'Play'. This loads the first file found.
6. Step through game. 'Shift' '3' initiates step mode. 'Space' steps through each move. 'Shift' '3' then ends mode.
7. Quit. 'Shift' 'Q' resets for new game.

```
1 REM <<<<<<<<< chess <> ISSI >>>>>>>>>
2 REM
10 CALL &BB03:SPEED WRITE 1
500 SYMBOL 240,1,3,29,63,63,57,28,7
510 SYMBOL 241,0,128,56,252,252,156,56,2
```

```
24
520 SYMBOL 242,7,7,4,31,16,127,64,127
530 SYMBOL 243,224,224,224,248,120,254,3
0,254
540 SYMBOL 244,1,19,81,91,109,55,27,15
550 SYMBOL 245,128,200,138,218,182,236,2
16,240
560 SYMBOL 246,1,1,7,14,28,28,14,7
570 SYMBOL 247,128,128,224,240,120,120,2
40,224
580 SYMBOL 248,0,0,3,7,15,31,9,3
590 SYMBOL 249,0,192,64,224,176,208,232,
120
600 SYMBOL 250,0,0,25,25,25,31,12,15
610 SYMBOL 251,0,0,152,152,152,248,240,2
40
620 SYMBOL 252,0,0,0,0,0,3,7,7
630 SYMBOL 253,0,0,0,0,0,192,224,224
640 SYMBOL 254,5,4,3,15,8,31,16,31
650 SYMBOL 255,224,224,192,240,112,248,5
6,248
660 PRINT CHR$(23);CHR$(1);
1000 CLEAR:GOSUB 21000
1010 FOR n=1 TO 100
1020 z$(n,1)="  "
1030 z$(n,2)="  "
1040 NEXT n
1100 GOSUB 20000
1110 LOCATE 27,18:PRINT "DO YOU WISH TO"
1115 LOCATE 27,20:PRINT "SET A BOARD ?"
1120 a$=INKEY$:a$=UPPER$(a$)
1130 IF a$="Y" THEN GOSUB 30000:IF check
=0 THEN GOTO 1170 ELSE GOTO 1100
1140 IF a$<>"N" THEN GOTO 1120
1150 GOSUB 22000:GOSUB 23000
1170 LOCATE 27,18:PRINT SPC(14);
1180 LOCATE 27,20:PRINT SPC(14);
1190 PRINT CHR$(23);CHR$(1);
1300 count=1
1500 GOSUB 29000
2000 GOTO 1500
20000 INK 0,0:INK 1,26:INK 2,6:INK 3,3
20290 PRINT #3," CHESS DUEL."
20300 PRINT #3,"-------------"
20310 LOCATE #3,2,6:PRINT #3,"FROM :-"
```

```
20320 LOCATE #3,2,9:PRINT #3,"TO    :-"
20500 RETURN
20980 REM >>>>>>>> initialize <<<<<<<<<
21000 DIM m$(8,8):DIM x$(8,8)
21010 DIM z$(100,2)
21020 k$=CHR$(240)+CHR$(241)
21030 k1$=CHR$(242)+CHR$(243)
21040 q$=CHR$(244)+CHR$(245)
21050 q1$=CHR$(242)+CHR$(243)
21060 b$=CHR$(246)+CHR$(247)
21070 b1$=CHR$(242)+CHR$(243)
21080 t$=CHR$(248)+CHR$(249)
21090 t1$=CHR$(242)+CHR$(243)
21100 r$=CHR$(250)+CHR$(251)
21110 r1$=CHR$(242)+CHR$(243)
21120 p$=CHR$(252)+CHR$(253)
21130 p1$=CHR$(254)+CHR$(255)
21250 RETURN
21980 REM >>>>>>>> game start <<<<<<<<<
22000 FOR n=1 TO 8
22010 FOR m=1 TO 8
22020 m$(n,m)=" "
22025 x$(n,m)=" "
22030 NEXT m
22040 NEXT n
22050 FOR n=1 TO 8
22060 m$(2,n)="P":x$(2,n)="P"
22070 m$(7,n)="p":x$(7,n)="p"
22080 NEXT n
22090 RESTORE 22250
22100 FOR n=1 TO 8
22110 READ a$
22120 m$(1,n)=a$
22125 x$(1,n)=a$
22130 a$=LOWER$(a$)
22140 m$(8,n)=a$
22145 x$(8,n)=a$
22150 NEXT n
22160 FOR n=1 TO 100
22170 z$(n,1)="  ":z$(n,2)="  "
22180 NEXT n:RETURN
22250 DATA R,T,B,Q,K,B,T,R
22980 REM >>>>>>>> draw pieces <<<<<<<<
23000 x=25:y=390
23010 FOR n=1 TO 8
```

```
23020 FOR m=1 TO 8
23030 c$=m$(n,m)
23040 IF c$=" " THEN GOTO 23100
23050 IF c$<"a" THEN col=2 ELSE col=3
23060 IF TEST(x-2,y+2)=3 THEN col=col XO
R 1
23070 PLOT 0,0,col
23080 MOVE x,y
23090 GOSUB 23500
23100 PLOT 0,0,0
23110 x=x+48
23130 NEXT m
23140 x=25:y=y-48
23150 NEXT n
23250 RETURN
23490 REM * * * piece * * *
23500 c$=UPPER$(c$)
23510 IF c$="K" THEN d$=k$:d1$=k1$
23520 IF c$="Q" THEN d$=q$:d1$=q1$
23530 IF c$="B" THEN d$=b$:d1$=b1$
23540 IF c$="T" THEN d$=t$:d1$=t1$
23550 IF c$="R" THEN d$=r$:d1$=r1$
23560 IF c$="P" THEN d$=p$:d1$=p1$
23600 PRINT #2,d$;
23610 MOVE x,y-16
23620 PRINT #2,d1$;
23750 RETURN
24055 IF cas=1 THEN RETURN
24500 n$="":cas=0
24510 a$=INKEY$
24520 IF a$="Q" THEN GOTO 1000
24522 IF a$="C" THEN GOSUB 26500
24524 IF cas=1 THEN RETURN
24525 IF a$="S" THEN GOSUB 31000
24530 IF a$="L" THEN GOSUB 32000
24535 IF a$="#" THEN z$(count,1)="FF":GO
TO 35000
24540 IF a$="R" THEN GOTO 25500
24545 a$=UPPER$(a$)
24550 IF a$<"A" OR a$>"H" THEN GOTO 2451
0
24555 n$=a$:SOUND 2,150,5,15:PRINT #3,n$
;
24560 a$=INKEY$
24570 IF a$<"1" OR a$>"9" THEN GOTO 2456
```

```
0
24580 n$=n$+a$:SOUND 2,150,5,15:PRINT #3
,a$;
24590 RETURN
24980 REM >>>>>>>>>>>> move <<<<<<<<<<<
25000 GOSUB 26000
25080 SOUND 2,300,75,15
25090 ro=s1:co=s:GOSUB 25200
25100 m$(s1,s)=m$(f1,f)
25110 IF play=1 AND f1=1 AND m$(f1,f)="p
" THEN m$(f1,f)="q"
25120 IF play=2 AND f1=8 AND m$(f1,f)="P
" THEN m$(f1,f)="Q"
25130 ro=s1:co=s:GOSUB 25200
25140 SOUND 2,100,20,15
25150 ro=f1:co=f:GOSUB 25200
25160 m$(f1,f)=" "
25170 RETURN
25200 x=25:y=390
25210 FOR n=1 TO 8
25220 FOR m=1 TO 8
25230 IF n=ro AND m=co THEN GOSUB 25300
25240 x=x+48
25250 NEXT m
25260 x=25:y=y-48
25270 NEXT n
25280 RETURN
25300 c$=m$(n,m)
25310 IF c$=" " THEN RETURN
25320 IF c$<"a" THEN col=2 ELSE col=3
25330 IF TEST(x-2,y+2)=3 THEN col=col XO
R 1
25340 PLOT 0,0,col:MOVE x,y
25350 GOSUB 23500
25360 RETURN
25480 REM
25490 REM * * * remove * * *
25500 SOUND 2,300,40,15:SOUND 2,400,50,1
5
25510 WINDOW #4,35,40,22,24
25515 PAPER #4,0:PEN #4,1
25520 INPUT #4,"Piece ",f$
25525 f$=UPPER$(f$)
25530 s$="A1":GOSUB 26000
25540 GOSUB 23000
```

```
25550 m$(f1,f)=" "
25560 z$(count,1)=f$:z$(count,2)=f$
25570 count=count+1
25580 IF count=101 THEN count=1
25590 CLS #4
25600 GOSUB 23000
25610 IF play=1 THEN play=2:GOSUB 28000
25620 GOTO 1500
25980 REM >>>>>>>>>> convert <<<<<<<<<<
26000 f=ASC(LEFT$(f$,1))
26010 f=f-64
26020 f1=VAL(RIGHT$(f$,1))
26030 f1=9-f1
26040 s=ASC(LEFT$(s$,1))
26050 s=s-64
26060 s1=VAL(RIGHT$(s$,1))
26070 s1=9-s1
26080 RETURN
26100 f$=CHR$(f+64)+HEX$(9-f1,1)
26130 s$=CHR$(s+64)+HEX$(9-s1,1)
26150 z$(count,1)=f$
26160 z$(count,2)=s$
26170 count=count+1
26180 RETURN
26480 REM >>>>>>>>>> castling <<<<<<<<<
26500 PRINT #3,"C";
26505 SOUND 2,600,25,15:SOUND 2,400,25,1
5:SOUND 2,200,25,15
26510 a$=INKEY$:a$=UPPER$(a$)
26520 IF a$<>"L" AND a$<>"R" THEN GOTO 2
6510
26530 PRINT #3,a$
26535 n$="C"+a$:cas=1:RETURN
26540 s$="A8":a$=RIGHT$(f$,1):GOSUB 2600
0:check=1
26545 IF (play=1 AND a$="L" AND m$(8,1)<
>"r") OR (play=1 AND a$="R" AND m$(8,8)<
>"r") THEN RETURN
26550 IF (play=2 AND a$="L" AND m$(1,1)<
>"R") OR (play=2 AND a$="R" AND m$(1,8)<
>"R") THEN RETURN
26560 IF play=1 AND m$(8,5)<>"k" THEN RE
TURN
26570 IF play=2 AND m$(1,5)<>"K" THEN RE
TURN
```

```
26575 au=1:en=7:IF play=1 THEN lin=8 ELS
E lin=1
26580 IF a$="L" THEN au=-1:en=2
26590 FOR n=5+au TO en STEP au
26600 IF m$(lin,n)<>" " THEN n=en:RETURN
26610 NEXT n
26620 f1=lin:f=5:s1=f1:s=5+2*au
26630 GOSUB 23000
26640 m$(f1,f)=" ":m$(s1,s)="k"
26645 GOSUB 26100
26650 IF lin=1 THEN m$(s1,s)="K"
26655 z$(count,1)="CC":z$(count,2)="CC":
count=count+1
26660 f1=lin:f=en+au:s1=f1:s=5+au
26670 m$(f1,f)=" ":m$(s1,s)="r"
26675 GOSUB 26100
26680 IF lin=1 THEN m$(s1,s)="R"
26685 GOSUB 23000
26690 check=0:RETURN
26980 REM >>>>>>>>>> validate <<<<<<<<<
27000 GOSUB 26000
27010 check=1
27020 c$=m$(f1,f):d$=m$(s1,s)
27030 IF c$=" " THEN RETURN
27040 IF c$>"Z" AND play=2 THEN RETURN
27050 IF c$<"a" AND play=1 THEN RETURN
27055 IF d$=" " THEN GOTO 27100
27060 IF d$>"Z" AND play=1 THEN RETURN
27070 IF d$<"a" AND play=2 THEN RETURN
27100 c$=UPPER$(c$):d$=UPPER$(d$)
27110 IF d$="K" THEN RETURN
27150 IF c$="P" THEN GOTO 27300
27160 IF c$="R" THEN GOTO 27400
27170 IF c$="T" THEN GOTO 27500
27180 IF c$="B" THEN GOTO 27600
27190 IF c$="Q" THEN GOTO 27700
27200 IF c$="K" THEN GOTO 27800
27230 check=0
27240 IF c$="K" OR c$="P" OR c$="T" THEN
 RETURN
27250 GOSUB 27900
27260 RETURN
27290 REM * * * pawn * * *
27300 IF play=1 AND s1>f1 THEN RETURN
27310 IF play=2 AND f1>s1 THEN RETURN
```

```
27320 dis=SGN(s1-f1)*(s1-f1):IF dis=1 TH
EN GOTO 27375
27330 IF dis>2 THEN RETURN
27340 IF s<>f THEN RETURN
27350 IF play=2 AND f1=2 AND m$(3,f)=" "
 THEN GOTO 27385
27360 IF play=1 AND f1=7 AND m$(6,f)=" "
 THEN GOTO 27385
27370 RETURN
27375 IF s=f AND d$=" " THEN GOTO 27385
27377 IF SGN(s-f)*(s-f)=1 AND d$<>" " TH
EN GOTO 27385
27380 RETURN
27385 GOTO 27230
27390 REM * * * rook * * *
27400 IF f1<>s1 AND f<>s THEN RETURN
27485 GOTO 27230
27490 REM * * * knight * * *
27500 IF s1=f1+2 AND s=f-1 THEN GOTO 275
80
27510 IF s1=f1+2 AND s=f+1 THEN GOTO 275
80
27520 IF s1=f1-2 AND s=f-1 THEN GOTO 275
80
27530 IF s1=f1-2 AND s=f+1 THEN GOTO 275
80
27540 IF s1=f1+1 AND s=f-2 THEN GOTO 275
80
27550 IF s1=f1+1 AND s=f+2 THEN GOTO 275
80
27560 IF s1=f1-1 AND s=f-2 THEN GOTO 275
80
27570 IF s1=f1-1 AND s=f+2 THEN GOTO 275
80
27575 RETURN
27580 GOTO 27230
27590 REM * * * bishop * * *
27600 xd=f-s:xd=SGN(xd)*xd
27610 yd=f1-s1:yd=SGN(yd)*yd
27620 IF xd<>yd THEN RETURN
27685 GOTO 27230
27690 REM * * * queen * * *
27700 xd=f-s:xd=SGN(xd)*xd
27710 yd=f1-s1:yd=SGN(yd)*yd
27720 IF xd=yd THEN GOTO 27785
```

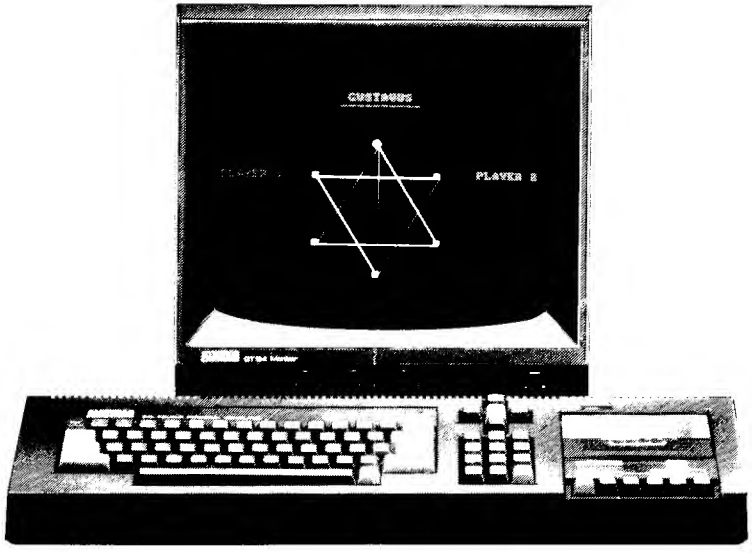
```
27730 IF f1<>s1 AND f<>s THEN RETURN
27785 GOTO 27230
27790 REM * * * king * * *
27800 IF SGN(f1-s1)*(f1-s1)<>1 AND SGN(f
-s)*(f-s)<>1 THEN RETURN
27885 GOTO 27230
27890 REM * * * check path * * *
27900 nd=0:md=0
27910 IF f1>s1 THEN nd=-1
27915 IF s1>f1 THEN nd=1
27920 IF f>s THEN md=-1
27925 IF s>f THEN md=1
27930 cn=f1:cm=f
27940 cn=cn+nd:cm=cm+md
27950 IF cn=s1 AND cm=s THEN RETURN
27960 IF m$(cn,cm)<>" " THEN check=1:RET
URN
27970 GOTO 27940
27980 REM >>>>>>>>>>> turn <<<<<<<<<<
28000 GOSUB 24000
28005 LOCATE #3,1,11:PRINT #3,"
   "
28010 IF cas=1 THEN GOSUB 26540 ELSE GOS
UB 27000
28020 IF check=0 THEN GOTO 28050
28030 LOCATE #3,1,11:PRINT #3,"INVALID M
OVE"
28040 SOUND 2,500,40,15,0,0,20:GOTO 2800
0
28050 IF cas=1 THEN GOTO 28090
28055 z$(count,1)=f$
28060 z$(count,2)=s$
28070 count=count+1:IF count=101 THEN co
unt=1
28080 GOSUB 25000
28090 RETURN
28980 REM >>>>>>> both players <<<<<<<
29000 play=1
29010 GOSUB 28000
29020 play=2
29030 GOSUB 28000
29040 RETURN
29980 REM >>>>>>>>> set board <<<<<<<<
30000 x=25:y=390
30010 FOR n=1 TO 8
```

```
30020 FOR m=1 TO 8
30030 SOUND 2,250,20,15
30040 m$(n,m)=" ":x$(n,m)=" "
30050 GOSUB 30500
30130 PLOT 0,0,0
30140 x=x+48
30150 NEXT m
30160 x=25:y=y-48
30170 NEXT n
30180 LOCATE 27,18:PRINT "IS THIS O.K. ?
"
30190 LOCATE 27,20:PRINT SPC(14)
30200 a$=INKEY$:a$=UPPER$(a$)
30210 IF a$="Y" THEN check=0:RETURN
30220 IF a$="N" THEN check=1:RETURN
30230 GOTO 30200
30490 REM * * * input * * *
30500 PLOT 0,0,2
30510 MOVE x,y:PRINT #2,"?";
30520 d$=INKEY$:c$=UPPER$(d$)
30530 IF d$="" THEN GOTO 30520
30540 PLOT 0,0,2
30550 MOVE x,y:PRINT #2,"?";
30560 IF c$<>"R" AND c$<>"N" AND c$<>"B"
 AND c$<>"Q" AND c$<>"K" AND c$<>"P" THE
N RETURN
30570 SOUND 2,100,20,15
30580 IF d$="n" THEN d$="t"
30590 IF d$="N" THEN d$="T"
30600 m$(n,m)=d$
30605 x$(n,m)=d$
30610 c$=d$
30620 IF c$<"a" THEN col=2 ELSE col=3
30630 IF TEST(x-2,y+2)=3 THEN col=col XO
R 1
30640 PLOT 0,0,col
30650 MOVE x,y
30660 GOSUB 23500
30670 RETURN
31040 g$=g$+m$(n,m)+x$(n,m)
31110 OPENOUT "!CHESS"
32010 OPENIN "!CHESS"
32050 m$(n,m)=LEFT$(g$,1):x$(n,m)=MID$(g
$,2,1)
32060 g$=RIGHT$(g$,LEN(g$)-2)
```

```
34980 REM >>>>>>>>>> replay <<<<<<<<<<
35000 count=0:play=1
35010 FOR n=1 TO 8:FOR m=1 TO 8
35015 m$(n,m)=x$(n,m)
35020 NEXT m:NEXT n
35025 GOSUB 20000:GOSUB 23000
35030 count=count+1
35040 IF z$(count,1)="FF" THEN GOTO 3550
0
35050 f$=z$(count,1):s$=z$(count,2)
35060 GOSUB 26000:GOSUB 25000
35070 count=count+1
35075 IF z$(count,1)="CC" THEN GOTO 3503
0
35080 a$=INKEY$
35090 IF a$="" THEN GOTO 35080
35095 play=2
35100 IF a$="#" THEN GOTO 35500
35110 IF z$(count,1)="FF" THEN GOTO 3550
0
35120 f$=z$(count,1):s$=z$(count,2)
35130 GOSUB 26000:GOSUB 25000
35135 IF z$(count+1,1)="CC" THEN count=c
ount+2:GOTO 35110
35140 a$=INKEY$
35150 IF a$="" THEN GOTO 35140
35155 play=1
35160 IF a$="#" THEN GOTO 35500
35170 GOTO 35030
35500 IF play=1 THEN GOTO 1500
35510 GOSUB 28000:GOTO 1500
```

Now 'MERGE' the 'BOARD CORE' routine.

9
Gustavus



A Novel Strategy Game for Two

As far as I am aware this is the first computer implementation of *Simple Simmons*, a game usually played with pencil and paper.

Simple Simmons or *Sim* was invented by a chap called Gustavus Simmons. We have decided to call this computer version of the game *Gustavus* as it sounds more grand than its paper cousin.

The game requires players alternately to draw a line between two points. The first to form a triangle of his own colour, whose points each rest on one of the original points, is the loser. The trick is to force your opponent into making the fatal triangle first.

The computer will take care of the turns together with the colour changes and will indicate the winner when a triangle is formed.

Lines are drawn by moving the flashing cross, clockwise or anti-clockwise, using either left or right cursor keys or joystick. When the cross is on the first point from which a line is required the copy key or fire button is pressed. A bleep signifies acceptance of the line origin. The cursor is next moved to the destination location, then as the copy key or fire button is pressed the line will be drawn. If an invalid line is

attempted a sound will indicate the fact. New start and finish points will then be required.

The game cannot end in a draw and there is no apparent advantage gained by being the first player. This game is very good for developing spacial awareness!

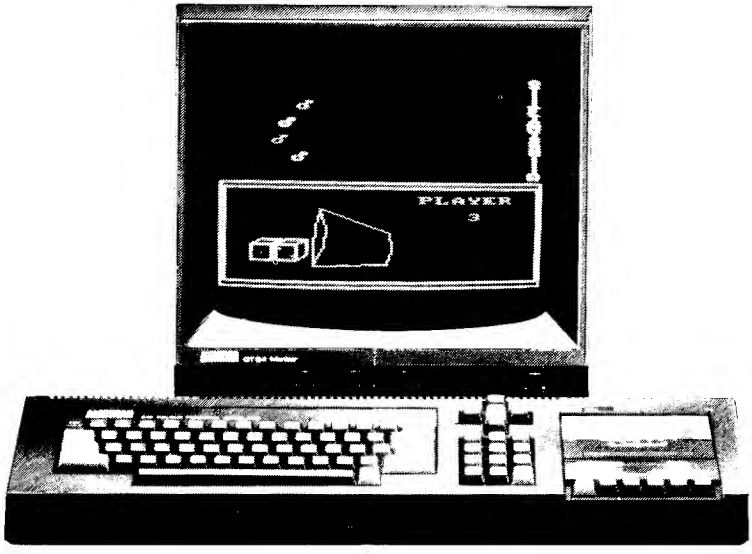
```
1 REM <<<<< gustavus <^> ISSI/JIM >>>>>
2 REM
1000 GOSUB 21000
1100 GOSUB 20000
1110 GOSUB 22000
1200 GOSUB 26000
1300 FOR n=1 TO 1000
1310 NEXT n
1390 IF play=1 THEN win=2 ELSE win=1
1400 LOCATE 14,24
1410 IF win=1 THEN PEN 2 ELSE PEN 3
1420 PRINT "PLAYER ";win;"won !"
1430 a$=INKEY$
1440 IF a$="" THEN GOTO 1430
1450 INK 1,26
1460 GOTO 1100
19980 REM >>>>>>>>>> screen <<<<<<<<<<
20000 INK 0,0:INK 1,26:INK 2,6:INK 3,20
20010 BORDER 0:PAPER 0:PEN 1:MODE 1
20020 LOCATE 17,1:PRINT "GUSTAVUS"
20030 LOCATE 16,2:PRINT "----------"
20040 FOR n=1 TO 6
20050 PLOT p(n,1),p(n,2),1
20060 DRAWR 0,8:DRAWR -8,0
20070 DRAWR 0,-8:DRAWR 8,0
20080 NEXT n
20090 LOCATE 1,10
20100 PEN 2
20110 PRINT "PLAYER 1"
20120 LOCATE 33,10
20130 PEN 3
20140 PRINT "PLAYER 2"
20150 TAG #3
20160 PRINT CHR$(23);CHR$(1);
20500 RETURN
20980 REM >>>>>>>>> initialize <<<<<<<<
21000 DIM p(6,2)
21010 RESTORE 21000
21020 FOR n=1 TO 6
```

```
21030 READ p(n,1)
21040 READ p(n,2)
21050 NEXT n
21060 DIM d(6,6)
21090 RETURN
21100 DATA 320,300,440,240,440,110
21110 DATA 320,50,200,110,200,240
21980 REM >>>>>>>>> start game <<<<<<<<<
22000 FOR n=1 TO 6
22010 FOR m=1 TO 6
22020 d(n,m)=0
22030 IF n=m THEN d(n,m)=3
22040 NEXT m
22050 NEXT n
22250 RETURN
22980 REM >>>>>>>>> joystick <<<<<<<<<<
23000 le=0:ri=0:fi=0
23010 a=JOY(0)
23020 b=JOY(1)
23030 a=a OR b
23040 IF (a AND 4)=4 THEN le=1
23050 IF (a AND 8)=8 THEN ri=1
23060 IF (a AND 16)=16 THEN fi=1
23070 a$=INKEY$
23080 IF a$=CHR$(242) THEN le=1
23090 IF a$=CHR$(243) THEN ri=1
23100 IF a$=CHR$(224) THEN fi=1
23120 FOR p=1 TO 50:NEXT p
23150 RETURN
23980 REM >>>>>>>>>>> move <<<<<<<<<<<
24000 point=1:IF play=1 THEN PLOT 0,0,2
ELSE PLOT 0,0,3
24010 MOVE p(point,1)-10,p(point,2)+10
24020 PRINT #3,CHR$(159);
24030 GOSUB 23000
24040 IF fi=1 THEN GOTO 24150
24050 MOVE p(point,1)-10,p(point,2)+10
24060 PRINT #3,CHR$(159);
24070 IF le=1 THEN point=point-1
24080 IF ri=1 THEN point=point+1
24090 IF point=7 THEN point=1
24100 IF point=0 THEN point=6
24120 GOTO 24010
24150 SOUND 2,200,50,15
24160 FOR w=1 TO 300
```

```
24170 b$=INKEY$
24180 NEXT w
24190 RETURN
24200 MOVE p(fro,1)-10,p(fro,2)+10
24210 PRINT #3,CHR$(159);
24220 MOVE p(des,1)-10,p(des,2)+10
24230 PRINT #3,CHR$(159);
24250 RETURN
24980 REM >>>>>>>>>>>> turn <<<<<<<<<<<
25000 GOSUB 24000
25010 fro=point
25020 GOSUB 24000
25030 des=point
25035 GOSUB 24200
25040 IF fro=des OR d(fro,des)<>0 THEN G
OTO 25200
25050 PLOT p(fro,1),p(fro,2)
25060 DRAW p(des,1),p(des,2)
25070 d(des,fro)=play
25080 d(fro,des)=play
25090 FOR n=1 TO 6
25110 IF d(des,n)=play AND d(n,fro)=play
 THEN dead=1:GOSUB 27000
25130 NEXT n
25190 RETURN
25200 SOUND 2,200,20,15:SOUND 2,300,20,1
5:SOUND 2,400,40,15
25210 GOTO 25000
25980 REM >>>>>>>>>>>> both <<<<<<<<<<<
26000 dead=0:INK 2,6,24
26010 play=1:GOSUB 25000
26020 INK 2,6
26030 IF dead=1 THEN RETURN
26040 INK 3,20,11
26050 play=2:GOSUB 25000
26060 IF dead=1 THEN RETURN
26070 INK 3,20
26080 GOTO 26000
26980 REM >>>>>>>>>> highlight <<<<<<<<
27000 PRINT CHR$(23);CHR$(0);
27005 INK 1,8,15:INK 2,6:INK 3,20
27010 PLOT p(fro,1),p(fro,2),1
27020 DRAW p(n,1),p(n,2)
27030 DRAW p(des,1),p(des,2)
27040 DRAW p(fro,1),p(fro,2)
```

```
27050 PRINT CHR$(23);CHR$(1);
27060 FOR s=200 TO 1 STEP-1
27070 SOUND 2,s,1,15
27080 NEXT s
27100 RETURN
```

10
Dice Derby



Shake!

This is the first of a dice trio to be followed by *Snake Eyes* and *Craps*. All three use the same central routines and a menu is provided to select between the games. This will produce an enjoyable set of dice games and save a lot of typing.

Dice Derby is a race game in which the dice total is used to determine how far each horse travels. It can be played by two to four players and has an attractive shaking cup feature.

The fire button is pressed to start the roll and again to finish. This means that each player can exercise control over the duration of each shake.

The characters will race along until one is declared the winner. The winner is not determined until the last throw of the round and so some very close 'photo finishes' are possible.

It is possible to end the game and return to the menu by pressing 'Shift' 'Q' for quit. Another game can then be selected.

Notes on typing in: the three dice games that follow can use the same routine called *Dicerama!* This should be typed first and then

saved onto tape. The games can then be added one by one. Of course it will not be possible to select a game from the menu if it has not yet been entered.

```
1 REM <<<<<<< dicerarma <> ISSI >>>>>>>
2 REM
100 ENV 1,1,-5,10,1,5,5
500 EVERY 500 GOSUB 5000:GOSUB 22000
1000 GOSUB 20000:PEN 2
1010 LOCATE 4,3:PRINT "1> Dice Derby"
1020 LOCATE 4,6:PEN 3:PRINT "2> Snake Ey
es":EI
1030 LOCATE 4,9:PEN 4:PRINT "3> Craps."
1050 a$=INKEY$:IF a$="" THEN GOTO 1050
1060 SOUND 2,200,25,15
1070 ON VAL(a$) GOTO 40000,50000,60000
1100 GOTO 1050
2000 PEN 8:LOCATE 4,10:PRINT "Press any
Key"
2010 a$=INKEY$:IF a$="" THEN GOTO 2010
2030 GOTO 1000
5000 DI:a$=INKEY$
5010 IF a$="Q" THEN RUN ELSE RETURN
19980 REM >>>>>>>>>> screen <<<<<<<<<<
20000 RESTORE 20000
20010 FOR n=0 TO 15
20020 READ a
20030 INK n,a
20040 NEXT n
20050 DATA 0,26,18,2,6,24,20,3
20060 DATA 8,0,0,0,0,0,0,0
20070 PAPER 0:PEN 1:BORDER 0:MODE 0
20080 PRINT CHR$(23);CHR$(0);
20090 PLOT 10,10,1:DRAW 629,10:DRAW 629,
190:DRAW 10,190:DRAW 10,10
20100 PLOT 0,0,1:DRAW 639,0:DRAW 639,200
:DRAW 0,200:DRAW 0,0
20110 PLOT 40,40,13:DRAW 80,30:DRAW 120,
30:DRAW 160,40
20120 DRAW 130,160:DRAW 100,155:DRAW 70,
160:DRAW 40,40
20130 PLOT 130,160:DRAW 100,165:DRAW 70,
160
20150 PLOT 50,45,12:DRAW 90,35:DRAW 130,
35:DRAW 170,45
```

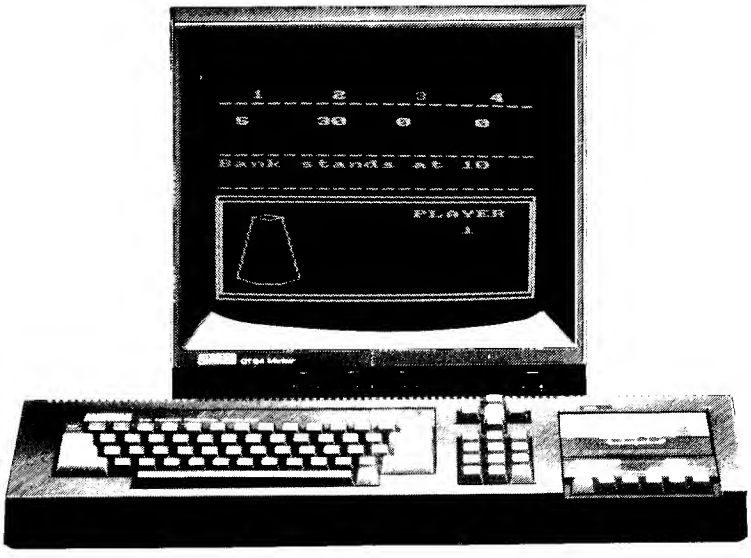
```
20160 DRAW 140,165:DRAW 110,160:DRAW 80,
165:DRAW 50,45
20170 PLOT 140,165:DRAW 110,170:DRAW 80,
165
20180 PLOT 180,40,11:DRAW 180,80:DRAW 18
5,110:DRAW 195,150
20190 DRAW 205,110:DRAW 200,80:DRAW 180,
40
20200 DRAW 325,45:DRAW 335,75:DRAW 330,1
05:DRAW 195,150
20210 PLOT 60,50,15:DRAW 60,80:DRAW 100,
80:DRAW 100,50:DRAW 60,50
20220 PLOT 110,50,15:DRAW 110,80:DRAW 15
0,80:DRAW 150,50:DRAW 110,50
20230 PLOT 60,80:DRAW 80,90:DRAW 120,90:
DRAW 100,80
20240 PLOT 110,80:DRAW 130,90:DRAW 170,9
0:DRAW 170,60:DRAW 150,50
20250 PLOT 170,90:DRAW 150,80
20500 RETURN
20980 REM >>>>>>>>>> rattle <<<<<<<<<<
21000 GOSUB 20090
21010 INK 11,0:INK 15,0
21020 INK 13,24:INK 12,0
21030 FOR q=1 TO 25:NEXT q
21040 SOUND 1,200,1,15
21050 INK 12,24:INK 13,0
21060 FOR q=1 TO 25:NEXT q
21070 SOUND 1,100,1,15
21160 d1=INT(RND(1)*6)+1
21170 d2=INT(RND(1)*6)+1
21190 a$=INKEY$:IF a$="" THEN 21010
21200 INK 13,0:INK 12,0
21210 INK 11,24:INK 15,20
21220 SOUND 2,150,25,15
21230 ch=d1:GOSUB 21400
21240 IF s(1)=1 THEN PLOT 70,72
21250 IF s(2)=1 THEN PLOT 70,65
21260 IF s(3)=1 THEN PLOT 70,58
21270 IF s(4)=1 THEN PLOT 95,72
21280 IF s(5)=1 THEN PLOT 95,65
21290 IF s(6)=1 THEN PLOT 95,58
21300 IF s(7)=1 THEN PLOT 82,65
21310 ch=d2:GOSUB 21400
21320 IF s(1)=1 THEN PLOT 118,72
```

```
21330 IF s(2)=1 THEN PLOT 118,65
21340 IF s(3)=1 THEN PLOT 118,58
21350 IF s(4)=1 THEN PLOT 140,72
21360 IF s(5)=1 THEN PLOT 140,65
21370 IF s(6)=1 THEN PLOT 140,58
21380 IF s(7)=1 THEN PLOT 128,65
21390 RETURN
21400 IF ch=0 THEN ch=1
21405 IF ch=1 THEN RESTORE 21500
21410 IF ch=2 THEN RESTORE 21510
21420 IF ch=3 THEN RESTORE 21520
21430 IF ch=4 THEN RESTORE 21530
21440 IF ch=5 THEN RESTORE 21540
21450 IF ch=6 THEN RESTORE 21550
21460 FOR n=1 TO 7
21470 READ s(n):NEXT n
21480 RETURN
21500 DATA 0,0,0,0,0,0,1
21510 DATA 1,0,0,0,0,1,0
21520 DATA 0,0,1,1,0,0,1
21530 DATA 1,0,1,1,0,1,0
21540 DATA 1,0,1,1,0,1,1
21550 DATA 1,1,1,1,1,1,0
21980 REM >>>>>>>> initialize <<<<<<<<<
22000 DIM s(7):DIM p(4,3):DIM o(4)
22050 RETURN
22480 REM >>>>>>>>>>> clear <<<<<<<<<<<
22500 LOCATE 1,1
22510 FOR n=1 TO 240
22520 PRINT " ";
22530 SOUND 2,241-n,1,15
22540 NEXT n
22580 LOCATE 1,1:RETURN
22980 REM >>>>>>>>>> players <<<<<<<<<<
23000 GOSUB 22500
23010 PRINT "PLAYERS ? (2-4)
23020 a$=INKEY$
23030 IF a$<"2" OR a$>"4" THEN 23020
23040 SOUND 2,300,25,15:SOUND 2,200,25,1
5:SOUND 2,100,25,15
23050 GOSUB 22500
23060 FOR n=1 TO 4
23070 p(n,1)=0
23080 NEXT n
23090 play=VAL(a$)
```

```
23100 FOR n=1 TO play
23110 o(n)=1
23120 NEXT n
23130 FOR q=n TO 4
23140 o(q)=0
23150 NEXT q
23250 RETURN
39980 REM >>>>>>>> dice derby <<<<<<<<<
40000 GOSUB 23000
40010 FOR n=1 TO 4
40020 p(n,1)=8:p(n,2)=384-n*32
40030 NEXT n
40035 PEN 6
40040 RESTORE 40100
40050 FOR n=1 TO 12
40052 IF n=4 THEN PEN 8
40054 IF n=10 THEN PEN 6
40060 READ a
40070 LOCATE 20,n
40080 PRINT CHR$(a);
40090 NEXT n
40100 DATA 235,149,149,70,73,78,73,83,72
,149,149,191
40150 GOSUB 49000
40200 FOR w=1 TO play
40204 PEN w+1
40210 LOCATE 13,15:PRINT "PLAYER";
40220 LOCATE 15,17:PRINT w
40230 SOUND 2,200,50,15
40240 a$=INKEY$
40250 IF a$="" THEN GOTO 40240
40260 PLOT 0,0,0:GOSUB 21230
40270 GOSUB 21000
40275 FOR d=1 TO (d1+d2)*2
40280 SOUND 2,400,1,15:PRINT CHR$(23);CH
R$(1);:GOSUB 49000
40290 p(w,1)=p(w,1)+4
40300 SOUND 2,200,1,15:GOSUB 49000:PRINT
 CHR$(23);CHR$(0);
40310 NEXT d
40350 NEXT w
40360 win=0
40370 FOR n=1 TO 4
40380 IF win>0 THEN GOTO 40400
40390 IF p(n,1)>570 THEN win=n
```

```
40400 NEXT n
40410 IF win=0 THEN GOTO 40200
40450 GOSUB 22500
40460 PRINT "  Player ";win;" won !"
40500 RESTORE 40600
40510 FOR n=1 TO 13
40520 READ a,b
40530 SOUND 2,a,b,15,1,0
40540 NEXT n
40590 GOTO 2000
40600 DATA 319,50,284,20,253,20,239,20
40610 DATA 253,20,284,20,319,50,319,10
40620 DATA 284,20,253,20,239,20,253,20
40630 DATA 284,20
49000 FOR n=1 TO 4
49010 TAG #1
49020 PLOT 0,0,n+1
49030 MOVE p(n,1),p(n,2)
49040 PRINT #1,CHR$(234);
49050 TAGOFF
49060 NEXT n
49070 PLOT 0,0,1
49080 RETURN
```

11
Snake Eyes



Rattle!

This is the second of the three dice games. *Dice Derby* must be entered first and then *Snake Eyes* can be added. The instructions for throwing the dice are the same.

The game is for two to four players and includes on-screen betting. At the start the target pot is entered, which may be between £40 and £400. The bank starts at £0.

Each player then rolls the dice. If a double is thrown then the player wins the bank amount. If 'Snake Eyes' (two ones) is thrown then the player gets twice the bank total. The bank resets to zero when it pays out.

After each round the bank gets £5 if the target set is below £200 and £10 if the target is above £200.

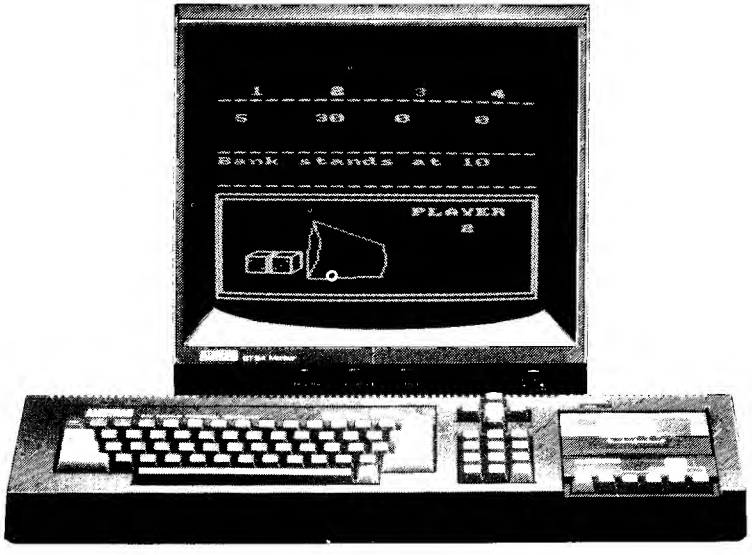
The game continues until a player's cash reaches the target score.

It may be interesting to change the amount to the bank after each round to provide a larger 'pot'!

```
49980 REM >>>>>>>> snake eyes <<<<<<<<<
50000 GOSUB 23000
50010 WINDOW #5,1,20,9,11:PEN #5,2
50020 PRINT " Enter Win Score ?"
50030 INPUT #5,lim:IF lim>400 OR lim<40
THEN GOTO 50000
50040 upda=5:GOSUB 22500:IF lim>200 THEN
 upda=10
50050 PEN 2:LOCATE 3,1:PRINT "1"
50060 PEN 3:LOCATE 8,1:PRINT "2"
50070 PEN 4:LOCATE 13,1:PRINT "3"
50080 PEN 5:LOCATE 18,1:PRINT "4"
50090 PEN 6:LOCATE 1,2:PRINT STRING$(20,
45);
50100 PEN 6:LOCATE 1,12:PRINT STRING$(20
,45);
50110 PEN 8:LOCATE 1,8:PRINT STRING$(20,
45);
50150 kit=0
50170 kit=kit+upda
50180 per=1
50190 FOR n=1 TO 4:LOCATE n*5-4,4
50195 PRINT p(n,1):NEXT n
50200 PEN per+1:LOCATE 13,15:PRINT "PLAY
ER"
50210 LOCATE 15,17:PRINT per
50220 GOSUB 50500:IF p(per,1)>lim THEN G
OTO 55000
50230 per=per+1:IF per>play THEN GOTO 50
170
50240 GOTO 50190
50500 a$=INKEY$
50510 IF a$="" THEN GOTO 50500
50540 CLS #5:PRINT #5,"Bank stands at";k
it
50550 PLOT 0,0,0:GOSUB 21230
50560 GOSUB 21000
50570 IF d1=1 AND d2=1 THEN GOSUB 50600
50580 IF d1=d2 AND d1<>1 THEN GOSUB 5070
0
50590 RETURN
50600 SOUND 2,300,25,15:SOUND 2,200,25,1
5
50610 SOUND 2,100,25,15:SOUND 2,300,25,1
5
```

```
50620 p(per,1)=p(per,1)+kit*2
50630 kit=0
50640 RETURN
50700 SOUND 2,400,25,15:SOUND 2,200,50,1
5
50710 SOUND 2,400,25,15:SOUND 2,100,50,1
5
50720 p(per,1)=p(per,1)+kit
50730 kit=0
50740 RETURN
55000 win=per
55010 GOTO 40450
```

12
Craps



... and Roll!

Here it is, the micro version of the American dice game featured in gangster movies and TV shows.

Two to four players can now pretend that they are in Las Vegas courtesy of the Amstrad.

At the start of the game each player has £100 (you can change it to dollars if you want more realism!). A bet is placed up to the total holding of each player on his turn. Then the fun begins.

If a 7 or 11 is thrown first then they are paid evens. If 2, 3 or 12 is thrown the turn ends and the stake is lost. Any other total becomes the *point*. This is then the total to roll for.

The player continues to roll until the point is reached, paying evens, or a 7 is thrown in which case it is the next player's turn.

If a player loses all his money then he drops out and the game continues until only one player is left.

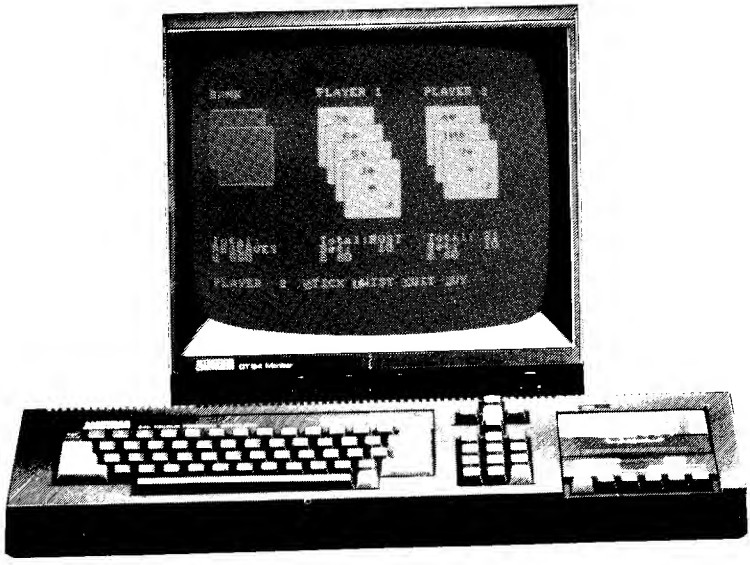
Pressing 'Shift' 'Q' will quit it any time.

This listing requires that the *Dice Derby* listing is entered first, since it uses the same start routines.

```
59980 REM >>>>>>>>>> craps <<<<<<<<<<
60000 GOSUB 23000
60010 PEN 2:LOCATE 3,1:PRINT "1"
60020 PEN 3:LOCATE 8,1:PRINT "2"
60030 PEN 4:LOCATE 13,1:PRINT "3"
60040 PEN 5:LOCATE 18,1:PRINT "4"
60050 PEN 6:LOCATE 1,2:PRINT STRING$(20,
45);
60060 PEN 6:LOCATE 1,12:PRINT STRING$(20
,45);
60065 PEN 8:LOCATE 1,8:PRINT STRING$(20,
45);
60070 WINDOW #5,1,20,9,11:PEN #5,2
60080 FOR n=1 TO 4
60090 p(n,1)=100
60100 NEXT n
60150 FOR w=1 TO play
60160 FOR n=1 TO 4
60170 LOCATE n*5-4,4:PRINT p(n,1)
60180 NEXT n
60185 IF p(w,1)<1 THEN GOTO 60500
60190 PRINT #5,"Player";w;"Bet ?"
60200 INPUT #5,bet:IF bet<1 OR bet>p(w,1
) THEN SOUND 2,300,50,15:GOTO 60200
60240 PLOT 0,0,0:GOSUB 21230
60250 GOSUB 21000
60260 tot=d1+d2
60270 IF tot=7 OR tot=11 THEN GOTO 61000

60275 IF tot=2 OR tot=3 OR tot=12 THEN G
OTO 62000
60280 point=tot
60290 PRINT #5,"  Roll again . . ."
60295 a$=INKEY$:IF a$="" THEN GOTO 60295
60297 PLOT 0,0,0:GOSUB 21230
60300 GOSUB 21000
60310 tot=d1+d2
60320 IF tot=point THEN GOTO 61000
60330 IF tot=7 THEN GOTO 62000
60340 GOTO 60290
60500 NEXT w
60505 flag=0:con=0
60510 FOR n=1 TO play
60520 IF con=1 THEN GOTO 60540
60530 IF p(n,1)>0 AND flag=1 THEN con=1
```

```
60535 IF p(n,1)>0 THEN flag=1
60540 NEXT n
60550 IF con=1 THEN GOTO 60150
60560 win=0
60570 FOR n=1 TO play
60580 IF win<>0 THEN GOTO 60600
60590 IF p(n,1)>0 THEN win=n
60600 NEXT n
60610 GOTO 40450
61000 CLS #5:SOUND 2,300,25,15:SOUND 2,2
00,25,15:SOUND 2,100,40,15
61010 PRINT #5,"      You win !"
61020 p(w,1)=p(w,1)+bet
61030 GOTO 60500
62000 CLS #5:SOUND 2,100,25,15:SOUND 2,2
00,25,15:SOUND 2,400,40,15
62010 PRINT #5,"      You lose !"
62020 p(w,1)=p(w,1)-bet
62030 GOTO 60500
```

13
Pontoon



A Pair can Break the Bank

In this computer version of the well known card game the bank starts with £500 and each player has £100. One or two players each play against the bank and aim to get a total of 21, which is Pontoon.

One card is first dealt face up to each player and one face down to the bank. A bet up to the total cash available may then be placed. The next card is dealt and various options are given to each player on their turn.

'S'tick. If the total is 16 or over then pressing 'S' will pass play on.
'T'wist. If another card is required this option may be taken by pressing 'T'.
'B'uy. Used when a hand looks good, to double the stake.

An ace will be taken to have the best value, i.e. 1 or 11 dependent upon the other cards. If at any time the card total exceeds 21 then play passes on. When both players have finished, the bank will then play its own hand to determine payout. It should be remembered that play is between the players and the bank, not against each other.

Wins and losses are calculated as follows: highest score is Pontoon with two cards, e.g. 10 and a Queen. The next highest score is a 5-*card trick*, which may be any five cards which do not exceed 21 in total. Pontoon with any card permutation is the third high score. If none, of the above exist then the highest score wins. Where the bank's total equals that of a player then the bank wins.

The game continues until each player has lost all available cash or until the bank is bust. 'Q' for Quit may be used to end a game prematurely.

This program shares many lines with the following card games. The routines common to all should be entered first, saved and then the remainder entered.

'CARD CORE'

```
9990 REM >>>> GENERATE PACK <<<<<<<<<<
10000 FOR suit=1 TO 4
10005 FOR car=1 TO 13
10010 pac$(suit,car)=CHR$(suit+225)
10015 IF car<11 AND car>1 THEN pac$(suit
,car)=pac$(suit,car)+STR$(car)
10016 IF car=1 THEN pac$(suit,car)=pac$(
suit,car)+" A"
10020 IF car=11 THEN pac$(suit,car)=pac$
(suit,car)+" J"
10025 IF car=12 THEN pac$(suit,car)=pac$
(suit,car)+" Q"
10030 IF car=13 THEN pac$(suit,car)=pac$
(suit,car)+" K"
10035 NEXT car
10040 NEXT suit
10045 RETURN
10490 REM >>>>>> SHUFFLE PACK <<<<<<<<
10500 FOR n=1 TO 52
10505 pa(n)=n:av(n)=1
10510 NEXT n
10515 FOR a=1 TO 30
10520 n1=INT(RND(1)*51)+1
10525 n2=INT(RND(1)*51)+1
10530 cc=pa(n1)
10535 pa(n1)=pa(n2)
10540 pa(n2)=cc
10545 NEXT a
10550 RETURN
```

```
10990 REM >>>>>>>> DRAW CARD <<<<<<<<<
10995 PRINT CHR$(24);
11000 LOCATE cx-1,cy
11005 PAPER 0:PEN 1:PRINT CHR$(230);:FOR
 z=1 TO 5:PRINT CHR$(224);:NEXT z:PRINT
CHR$(231)
11010 LOCATE cx-1,cy+1
11020 PRINT CHR$(234);:PAPER 1:PEN 0:PRI
NT RIGHT$(pac$(suit,car),2);
11021 IF LEFT$(pac$(suit,car),1)=CHR$(22
7) OR LEFT$(pac$(suit,car),1)=CHR$(228)
THEN PAPER 1:PEN 2 ELSE PAPER 1:PEN 0

11022 PRINT LEFT$(pac$(suit,car),1);:PAP
ER 0:PEN 1:PRINT CHR$(143);CHR$(143);CHR
$(235)
11025 LOCATE cx-1,cy+2
11030 PRINT CHR$(234);:FOR z=1 TO 5:PRIN
T CHR$(143);:NEXT z:PRINT CHR$(235)
11035 LOCATE cx-1,cy+3
11040 PRINT CHR$(234);CHR$(143);CHR$(143
);:IF LEFT$(pac$(suit,car),1)=CHR$(227)
OR LEFT$(pac$(suit,car),1)=CHR$(228) THE
N PAPER 1:PEN 2:ELSE PAPER 1:PEN 0
11045 PRINT LEFT$(pac$(suit,car),1);:PEN
 1:PAPER 0:PRINT CHR$(143);CHR$(143);CHR
$(235)
11050 LOCATE cx-1,cy+4
11055 PRINT CHR$(234);:FOR z=1 TO 5:PRIN
T CHR$(143);:NEXT z:PRINT CHR$(235)
11060 LOCATE cx-1,cy+5
11065 PRINT CHR$(234);CHR$(143);CHR$(143
);CHR$(143);:PAPER 1:PEN 0:PRINT RIGHT$(
pac$(suit,car),2);:PAPER 0:PEN 1:PRINT C
HR$(235)
11070 LOCATE cx-1,cy+6
11075 PRINT CHR$(233);:FOR z=1 TO 5:PRIN
T CHR$(225);:NEXT z:PRINT CHR$(232)
11076 SOUND 1,100,2
11080 RETURN
11170 REM >>>>> DRAW CARD BACK <<<<<
11200 LOCATE cx-1,cy:PEN 3
11205 PRINT CHR$(236);:FOR n=1 TO 5:PRIN
T CHR$(240);:NEXT n:PRINT CHR$(237)
11210 FOR nn=1 TO 5:LOCATE cx-1,cy+nn
```

```
11215 PRINT CHR$(242);:FOR n=1 TO 5:PRIN
T CHR$(244);:NEXT n:PRINT CHR$(243)
11220 NEXT nn
11225 LOCATE cx-1,cy+6:PRINT CHR$(239);:
FOR n=1 TO 5:PRINT CHR$(241);:NEXT n:PRI
NT CHR$(238)
11226 SOUND 1,300,2
11230 PEN 1:RETURN
16990 REM >>>>>>>> CHARACTERS <<<<<<<<<
17000 SYMBOL 224,0,255,255,255,255,255,2
55,255
17005 SYMBOL 225,255,255,255,255,255,255
,255,0
17010 SYMBOL 230,0,63,127,127,127,127,12
7,127
17015 SYMBOL 231,0,252,254,254,254,254,2
54,254
17020 SYMBOL 232,254,254,254,254,254,254
,252,0
17025 SYMBOL 233,127,127,127,127,127,127
,63,0
17030 SYMBOL 234,127,127,127,127,127,127
,127,127
17035 SYMBOL 235,254,254,254,254,254,254
,254,254
17040 SYMBOL 236,0,63,106,85,106,85,106,
85
17045 SYMBOL 237,0,252,170,86,170,86,170
,86
17050 SYMBOL 238,170,86,170,86,170,86,25
2,0
17055 SYMBOL 239,106,85,106,85,106,85,63
,0
17060 SYMBOL 240,0,255,170,85,170,85,170
,85
17065 SYMBOL 241,170,85,170,85,170,85,25
5,0
17070 SYMBOL 242,106,85,106,85,106,85,10
6,85
17075 SYMBOL 243,170,86,170,86,170,86,17
0,86
17080 SYMBOL 244,170,85,170,85,170,85,17
0,85
```

'Card core' should now be saved separately for use in other card games.

```
1 REM << PONTOON            ANDY / JIM >>>
2 DEFINT a-z
3 DIM pa(52),av(52),pac$(5,13),cc(3)
4 DIM cx(4),cy(4)
5 DIM noc(4),pl(2)
6 DIM plc(3,5),bc(5)
7 DIM mo(2),p$(2)
8 SYMBOL AFTER 223
9 GOSUB 17000
10 GOSUB 11500
11 mo(1)=100:mo(2)=100:mc=500
12 cc(1)=20:cc(2)=33:cc(3)=6
13 pl(1)=1:pl(2)=1
15 GOSUB 10000
20 GOSUB 10500
21 CLS:IF np=1 THEN pl(2)=0
22 IF pl(1)=0 AND pl(2)=0 THEN 22000
23 IF pl(1)=0 THEN be(1)=0
24 IF pl(2)=0 THEN be(2)=0
25 GOSUB 12000
30 cx(1)=15:cx(2)=28
35 cy(1)=5:cy(2)=5
40 bx=2:by=5
42 GOSUB 13500
45 FOR q=1 TO np
46 IF pl(q)=0 THEN 55
50 GOSUB 14000
51 GOSUB 18000
55 NEXT q
56 cx=bx:cy=by:FOR q=1 TO nc
57 GOSUB 11200:cy=cy+2:cx=cx+1:NEXT q
60 GOSUB 16000
65 FOR q=1 TO np:IF pl(q)=0 THEN 80
66 cy=cy(q):cx=cx(q)
70 cy=cy+2:cx=cx+1
75 GOSUB 16500
76 GOSUB 18000
77 cx(q)=cx%
80 NEXT q
81 cx=bx+1:cy=by+2:GOSUB 11100
85 GOSUB 11200
90 GOSUB 14500
91 REM >>>>>>>> SCORE LOGIC <<<<<<<<<<
92 q=3
```

```
93 IF np=2 THEN 94 ELSE IF sc(1)>21 THEN
 p$(1)="L":GOTO 150
94 IF sc(1)>21 AND sc(2)>21 THEN p$(1)="
L":p$(2)="L":GOTO 150
95 FOR w=1 TO nc
100 cx=bx:cy=(by-2)+w*2
105 plc(3,w)=bc(w)
110 GOSUB 13000:GOSUB 11000
111 bx=bx+1
115 NEXT w
116 bx=cx:by=cy
120 FOR w=1 TO nc
125 plc(3,w)=bc(w):NEXT w
130 q=3:noc(3)=nc:cx(3)=bx:GOSUB 18000
135 IF sc(3)<16 THEN GOSUB 11100 ELSE GO
TO 150
136 bx=bx+1:by=by+2:plc(3,nc)=bc(nc):q=3
:cx=bx:cy=by:w=nc:GOSUB 13000:GOSUB 1100
0
137 noc(3)=nc:GOSUB 18000
140 IF nc=5 THEN 150 ELSE GOTO 135
145 f=(RND(1)*2)+17:IF sc(3)<f THEN GOSU
B 11100:GOTO 136
150 q=1
151 IF pl(q)=0 THEN 200
152 IF sc(q)>21 THEN p$(q)="L":GOTO 200
153 IF sc(3)>21 THEN p$(q)="W":GOTO 200
160 IF nc=5 AND noc(q)=5 THEN p$(q)="L":
GOTO 200
162 IF noc(q)=5 THEN p$(q)="W":GOTO 200
163 IF nc=5 THEN p$(q)="L":GOTO 200
164 IF sc(3)>sc(q) THEN p$(q)="L":GOTO 2
00
165 IF sc(3)=21 AND nc=2 AND sc(q)=21 AN
D noc(q)=2 THEN p$(q)="L":GOTO 200
170 IF noc(q)=5 AND nc<5 THEN p$(q)="W":
GOTO 200
175 IF sc(q)=21 AND noc(q)=2 THEN p$(q)=
"W":GOTO 200
180 IF sc(q)>sc(3) THEN p$(q)="W":GOTO 2
00
185 IF sc(q)=sc(3) THEN p$(q)="L":GOTO 2
00
186 IF sc(3)>sc(q) THEN p$(q)="L"
200 IF np=2 AND q=1 THEN q=2:GOTO 151
```

```
201 IF np=1 THEN 225
205 IF p$(1)="L" AND p$(2)="L" THEN 550
210 IF p$(1)="W" AND p$(2)="W" THEN 500
215 IF p$(1)="W" AND p$(2)="L" THEN 600
220 IF p$(1)="L" AND p$(2)="W" THEN 650
225 IF p$(1)="W" THEN 400
230 IF p$(1)="L" THEN 450
240 END
395 REM >>>>>>>>> PLAYER WIN <<<<<<<<<
400 mc=mc-be(1)
405 mo(1)=mo(1)+be(1)*2
406 GOSUB 23000
410 IF mc<1 THEN 750
411 GOSUB 19000
412 LOCATE 1,25:PRINT"Player wins":GOSUB
 19500
415 GOTO 20
445 REM >>>>>>>>>> BANK WIN <<<<<<<<<<
450 mc=mc+be(1)
451 GOSUB 19000:GOSUB 23500
452 LOCATE 1,25:PRINT"Bank wins":GOSUB 1
9500
453 IF mo(1)<1 THEN pl(1)=0
455 GOTO 20
495 REM >>>>>>>>> BOTH WIN <<<<<<<<<<<
500 mo(1)=mo(1)+be(1)*2
505 mo(2)=mo(2)+be(2)*2
506 GOSUB 23000
510 mc=mc-be(1)-be(2)
515 IF mc<1 THEN 750
516 GOSUB 19000
520 GOTO 20
545 REM >>>>>>>>> BOTH LOST <<<<<<<<<<<
550 mc=mc+be(1)+be(2)
551 GOSUB 19000:GOSUB 23500
553 IF mo(1)<1 THEN pl(1)=0
554 IF mo(2)<1 THEN pl(2)=0
555 IF pl(1)=0 AND pl(2)=1 THEN LOCATE 1
,25:PRINT "Player 2 loses"
560 IF pl(1)=1 AND pl(2)=0 THEN LOCATE 1
,25:PRINT "Player 1 loses"
565 IF pl(1)=1 AND pl(2)=1 THEN LOCATE 1
,25:PRINT "Both players lose"
566 GOSUB 19500
570 GOTO 20
```

```
595 REM >>>>>>>>> PLAYER 1 WIN <<<<<<<<<
600 mc=mc+be(2)
605 mo(1)=mo(1)+be(1)*2
610 mc=mc-be(1)
611 GOSUB 23000
614 IF mo(2)<1 THEN pl(2)=0
615 IF mc<1 THEN 750
616 GOSUB 19000
617 LOCATE 1,25:PRINT"Player 1 wins":GOS
UB 19500
620 GOTO 20
645 REM >>>>>>> PLAYER 2 WIN <<<<<<<<<<
650 mc=mc+be(1)
655 mo(2)=mo(2)+be(2)*2
660 mc=mc-be(2)
661 GOSUB 23000
664 IF mo(1)<1 THEN pl(1)=0
665 IF mc<1 THEN 750
666 GOSUB 19000
667 LOCATE 1,25:PRINT"Player 2 wins":GOS
UB 19500
670 GOTO 20
745 REM >>>>>>>> BANK BROKE <<<<<<<<<<<
750 GOSUB 19000
755 LOCATE 1,25
760 PRINT"Well done,you broke the bank"
761 GOSUB 23000:GOSUB 23000
765 GOSUB 19500
770 CLS
775 PRINT"PLAYER 1:";mo%(1)
780 IF np=2 THEN PRINT"PLAYER 2:";mo(2)
785 GOSUB 19500:GOSUB 19500
790 GOTO 10
11490 REM >>>>> NUMBER OF PLAYERS <<<<<
11500 MODE 1:WINDOW #3,24,40,25,25:INK 1
,26:INK 2,6:BORDER 0:INK 0,0:INK 3,14
11501 LOCATE 15,1:PRINT"PONTOON"
11505 PRINT:INPUT "1 or 2 players ";np
11510 IF np<>1 AND np<>2 THEN 11500
11515 FOR q=1 TO np:pl(q)=1:NEXT q
11520 RETURN
11990 REM >>>>>> DEAL CARDS <<<<<<
12000 FOR q=1 TO np
12005 c1=(RND(1)*51)+1
12015 IF av(c1)=0 THEN 12005
```

```
12020 av(c1)=0
12025 plc(q,1)=c1
12030 NEXT q
12035 c1=(RND(1)*51)+1
12045 IF av(c1)=0 THEN 12035
12050 av(c1)=0
12055 bc(1)=c1
12060 noc(1)=1:noc(2)=1:nc=1
12065 RETURN
12990 REM >>>>> DECODE CARD <<<<<<<
13000 sui=plc(q,w)/13:car=plc(q,w) MOD 1
3:IF car=0 THEN car=13
13005 IF sui>0 THEN suit=1
13010 IF sui>1 THEN suit=2
13015 IF sui>2 THEN suit=3
13020 IF sui>3 THEN suit=4
13025 RETURN
13490 REM >>>>>> DRAW SCREEN <<<<<<<
13500 LOCATE 1,3:PRINT "BANK","PLAYER 1"
,:IF np=2 THEN PRINT"PLAYER 2"
13505 LOCATE 14,21:PRINT"Bet:":IF np=2 T
HEN LOCATE 27,21:PRINT"Bet:"
13510 LOCATE 1,22:PRINT"£":LOCATE 14,22:
PRINT"£":LOCATE 27,22:PRINT"£"
13515 LOCATE 1,21:PRINT"RESERVES":LOCATE
 1,20:PRINT"Total:":LOCATE 14,20:PRINT"T
otal:":IF np=2 THEN LOCATE 27,20:PRINT"T
otal:"
13520 RETURN
13990 REM >>>>>>>> DRAW HAND <<<<<<<<<
14000 cx=cx(q):cy=cy(q)
14005 FOR w=1 TO noc(q)
14015 GOSUB 13000
14020 GOSUB 11000
14025 cy=cy+2:cx=cx+1:NEXT w
14030 LOCATE cc(q)-5,22:PRINT mo(q)
14035 RETURN
14490 REM >>>>>>> STICK <<<<<<<<
14500 FOR q=1 TO np:IF pl(q)=0 THEN 1452
6
14501 cx(q)=cx(q)+1
14505 LOCATE 1,25:FOR n=1 TO 39:PRINT "
";:NEXT n
14507 LOCATE 1,25:PRINT"PLAYER ";q;",";:
```

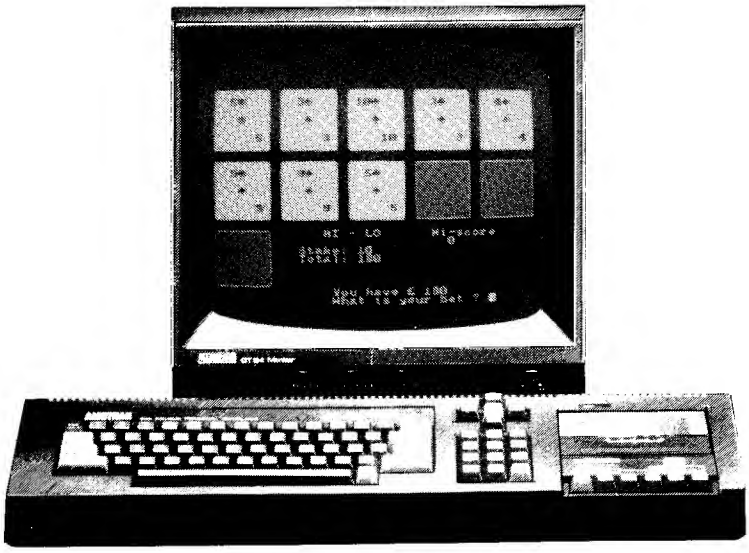
```
PRINT CHR$(24);"S";CHR$(24);"TICK ";CHR$
(24);"T";CHR$(24);"WIST";
14508 PRINT " ";CHR$(24);"Q";CHR$(24);"U
IT ";CHR$(24);"B";CHR$(24);"UY"
14510 a$=INKEY$:a$=UPPER$(a$):IF a$="S"
OR a$="T" OR a$="B" OR a$="Q" THEN 14511
 ELSE GOTO 14510
14511 cx=cx(q):cy=cy(q)+(2*noc(q))
14512 IF a$="B" THEN GOTO 20000
14513 IF a$="Q" THEN END
14515 IF a$="S" AND sc(q)>15 THEN GOSUB
15000:cx=cx+1:GOTO 14526
14520 IF a$="T" THEN GOSUB 15500:cx=cx+1
:cx(q)=cx(q)+1
14523 GOSUB 18000
14524 IF sc(q)>21 THEN LOCATE cc(q),20:P
RINT"BUST":GOTO 14526
14525 IF noc(q)<>5 THEN 14510
14526 NEXT q
14530 RETURN
14990 REM
15000 GOSUB 18000:LOCATE cc(q),20:PRINT"
   ":LOCATE cc(q),20:PRINT sc(q):RETURN
15490 REM >>>>>>>> TWIST <<<<<<<<<<
15500 c=(RND(1)*51)+1:IF av(c)=0 THEN 15
500:av(c)=0:noc(q)=noc(q)+1
15510 plc(q,noc(q))=c:w=noc(q):GOSUB 130
00:GOSUB 11000:RETURN
15990 REM >>>>>> PLACE BETS <<<<<<<<<
16000 LOCATE 2,22:PRINT mc:FOR q=1 TO np

16001 IF pl(q)=0 THEN 16020
16005 LOCATE 1,25:FOR n=1 TO 39:PRINT" "
;:NEXT n
16010 LOCATE 1,25:PRINT"PLAYER ";q;"Your
 bet (£) ";
16011 INPUT #3,be(q):IF be(q)<1 THEN 16
005
16012 IF be(q)>mo(q) THEN 16005
16015 LOCATE cc(q),21:PRINT be(q)
16016 LOCATE cc(q),22:PRINT"    "
16017 mo(q)=mo(q)-be(q)
16018 LOCATE cc(q)-4,22:PRINT"    ":LOCAT
E cc(q)-5,22:PRINT mo(q)
16020 NEXT q
```

```
16040 RETURN
16490 REM >>>> DEAL ANOTHER CARD <<<<
16500 c=(RND(1)*51)+1
16505 IF av(c)=0 THEN 16500
16510 plc(q,noc(q)+1)=c:av(c)=0
16515 noc(q)=noc(q)+1
16520 w=noc(q):GOSUB 13000:GOSUB 11000
16525 RETURN
17085 RETURN
17500 GOSUB 18000
17505 IF sc(q)=21 THEN END
17510 RETURN
17990 REM >>>>>> PLAYER'S SCORE <<<<<<<
18000 sc(q)=0:ac=0:FOR z=1 TO noc(q):w=z

18005 GOSUB 13000:IF RIGHT$(pac$(suit,ca
r),2)=" K" OR RIGHT$(pac$(suit,car),2)="
 Q" OR RIGHT$(pac$(suit,car),2)=" J" THE
N sc(q)=sc(q)+10
18010 IF RIGHT$(pac$(suit,car),2)=" A"TH
EN ac=ac+1:GOTO 18020
18015 sc(q)=sc(q)+VAL(RIGHT$(pac$(suit,c
ar),2))
18020 NEXT z:IF ac>0 THEN GOSUB 18200
18021 LOCATE cc(q),20:PRINT"    ":LOCATE
cc(q),20:PRINT sc(q)
18025 RETURN
18190 REM >>>>> ACE IN HAND <<<<<<<<<
18200 FOR r=1 TO ac
18202 IF sc(q)+11>21 THEN sc(q)=sc(q)+1
ELSE sc(q)=sc(q)+11
18204 NEXT r
18205 GOTO 18021
19000 LOCATE 1,25:FOR a=1 TO 39:PRINT" "
;:NEXT a:RETURN
19500 FOR t=1 TO 3000:NEXT t:RETURN
19990 REM >>>>>>>>> BUY <<<<<<<<<<<<<<
20000 IF mo(q)<be(q) THEN 14523
20004 be(q)=be(q)*2
20005 mo(q)=mo(q)-be(q)/2
20010 GOSUB 15500
20011 LOCATE cc(q),21:PRINT"    "
20012 LOCATE cc(q),21:PRINT be(q)
20013 LOCATE cc(q)-5,22:PRINT"    "
20014 LOCATE cc(q)-5,22:PRINT mo(q)
```

```
20020 cx=cx+1:cx(q)=cx(q)+1
20025 GOTO 14513
21990 REM >>>>> END OF GAME <<<<<<<<<<
22000 GOSUB 19000:LOCATE 1,25:PRINT"All
Player's money gone"
22001 FOR so=100 TO 500 STEP 5
22002 SOUND 1,so,2:NEXT so
22005 GOSUB 19500:GOTO 10
22990 REM >>>>>> WIN SOUND <<<<<<<<<<
23000 FOR so=300 TO 100 STEP -10
23005 SOUND 1,so,1
23010 NEXT so
23015 RETURN
23490 REM >>>>>> LOSE SOUND <<<<<<<<<
23500 SOUND 1,200,20
23505 SOUND 1,0,10
23510 SOUND 2,500,40
23515 RETURN
```

Now 'MERGE' the 'CARD CORE' routine. . .

14
Hi-Lo



Nothing for a Pair

Here is the computer version of the card game featured in a well known TV programme.

Each player starts with £150 and eleven cards are dealt out face down.

The first is turned over and a bet is requested. The bet can be anything up to the player's total holding. When the bet has been placed, 'higher' or 'lower' will be requested. If this is correct then the bet is paid evens. If the guess is wrong or the card is the same value then the stake for that card is lost. This then continues until all the cards have been turned and a new game can then be played.

The 'High Score' subroutine should be merged with this program to provide another competitive dimension or simply to record how well you perform from one game to the next.

Since this game shares many lines with the *Pontoon* listing, these have not been re-listed. To complete entry of the game the 'Card Core' should be loaded first and then the additional lines for *Hi-Lo*

can be entered. The 'Card Core' is also required for the next game, so do not rub over it!

```
1 REM <<< HI-LO          ANDY / JIM  >>>>
2 REM
3 DIM pa(52),av(52),pac$(5,13),bc(11)
4 DIM cx(11),cy(11)
5 hi=0:gam$="Hi-Lo":addr=34999
6 BORDER 0
8 SYMBOL AFTER 223
9 GOSUB 17000
10 MODE 1:INK 1,26:INK 2,6:INK 3,14:INK
0,0
11 WINDOW #3,32,40,25,25
12 PEN 1
15 LOCATE 14,17:PRINT"HI - LO"
20 GOSUB 10000
22 GOSUB 10500
25 GOSUB 11500
30 LOCATE 11,19:PRINT"Stake:"
35 LOCATE 11,20:PRINT"Total:"
37 LOCATE 27,17:PRINT"Hi-score"
40 tot=150
45 be=0
50 GOSUB 13500
60 GOSUB 14500
65 c=1
70 cx=cx(c):cy=cy(c)
71 GOSUB 13500
72 IF c=11 THEN 25000
75 plc(1,1)=bc(c):q=1:w=1
76 GOSUB 13000:GOSUB 11000
80 GOSUB 14000
85 GOSUB 15000
86 GOSUB 13500
90 cx=cx(c+1):cy=cy(c+1)
95 plc(1,1)=bc(c+1):ca=car
100 q=1:w=1:GOSUB 13000
101 GOSUB 11000
105 IF car>ca AND a$="H" THEN 16500:GOTO
 70
110 IF car<ca AND a$="L" THEN 16500:GOTO
 70
115 GOTO 16000
120 GOTO 70
```

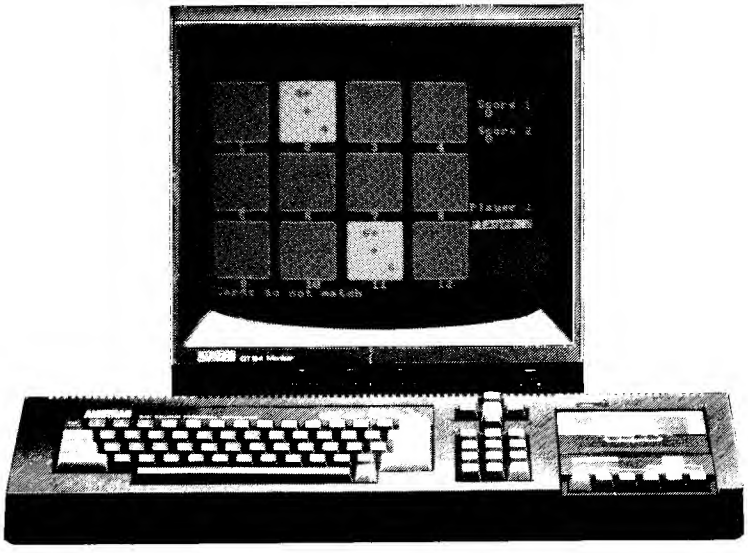
```
11490 REM >>>>>>> DRAW CARDS <<<<<<<<<
11500 RESTORE 11525:FOR a=1 TO 11
11505 READ cx(a),cy(a)
11506 cx=cx(a):cy=cy(a)
11510 GOSUB 11200
11515 NEXT a
11520 RETURN
11525 DATA 2,1,10,1,18,1,26,1,34,1
11530 DATA 2,9,10,9,18,9,26,9,34,9
11535 DATA 2,17
12990 REM >>>>>> DECODE CARD <<<<<<<<<
13000 sui=plc(q,w)/13:car=plc(q,w) MOD 1
3:IF car=0 THEN car=13
13005 IF sui>0 THEN suit=1
13010 IF sui>1 THEN suit=2
13015 IF sui>2 THEN suit=3
13020 IF sui>3 THEN suit=4
13025 RETURN
13490 REM >>> PRINT HI,TOTAL,BET <<<<<
13500 LOCATE 17,19:PRINT "     "
13501 LOCATE 17,19:PRINT be
13504 LOCATE 17,20:PRINT "       "
13505 LOCATE 17,20:PRINT tot
13509 LOCATE 28,18:PRINT "       "
13510 LOCATE 28,18:PRINT hi
13515 RETURN
13990 REM >>>>>>> PLACE BETS <<<<<<<<<
14000 LOCATE 15,24:PRINT"You have £";tot
14001 LOCATE 15,25:PRINT"What is your be
t "
14005 INPUT #3,be
14010 IF be<1 OR be>tot THEN 14005
14015 tot=tot-be:GOSUB 13500
14020 RETURN
14490 REM >>>> GENERATE CARDS <<<<<<<
14500 FOR a=1 TO 11
14505 c=INT(RND(1)*51)+1
14510 IF av(c)=0 THEN 14505
14515 bc(a)=pa(c)
14520 av(c)=0
14525 NEXT a
14530 RETURN
14990 REM >>>>>> HIGHER / LOWER <<<<<<
15000 LOCATE 15,25:FOR b=1 TO 25:PRINT"
```

```
";:NEXT b
15001 LOCATE 15,24:FOR a=1 TO 20:PRINT"
";:NEXT A
15005 LOCATE 15,25:PRINT CHR$(24);"H";CH
R$(24);"igher / ";CHR$(24);"L";CHR$(24);
"ower"
15010 a$=INKEY$:a$=UPPER$(a$)
15015 IF a$<>"H" AND a$<>"L" THEN 15010
15020 RETURN
15990 REM >>>>>>>> LOSE <<<<<<<<<<<<
16000 SOUND 1,300,20:SOUND 1,0,10
16010 SOUND 1,500,40
16020 IF tot<1 THEN 26000
16025 c=c+1:GOTO 70
16490 REM >>>>>>>>> WIN <<<<<<<<<<<<
16500 FOR so=500 TO 300 STEP -10
16505 SOUND 1,so,2:NEXT so
16510 tot=tot+be*2
16520 c=c+1:GOTO 70
17085 SYMBOL 226,56,56,254,254,254,16,56
,0
17090 SYMBOL 227,16,56,124,254,124,56,16
,0
17095 SYMBOL 228,108,254,254,254,124,56,
16,0
17100 SYMBOL 229,16,56,124,254,254,16,56
,0
17105 MEMORY 34999:GOSUB 29500
17110 RETURN
19500 FOR t=1 TO 3000:NEXT t:RETURN
24990 REM >>>>>> HIGH SCORES <<<<<<<<<
25000 IF tot>hi THEN hi=tot
25005 sc=tot:GOSUB 60000
25010 GOTO 10
25990 REM >>>>>>> NO MONEY LEFT <<<<<<
26000 LOCATE 15,25:FOR a=1 TO 25:PRINT"
";:NEXT a
26005 LOCATE 15,25:PRINT "Run out of mon
ey !"
26010 GOSUB 19500:GOTO 10
```

Now 'MERGE' the 'INKEYS' , the

'HIGH SCORE' and the 'CARD CORE'. . .

15
Concentration



Find the Pair

This deceptively simple game requires a good memory and of course great concentration.

Each player has one turn to select two cards. If they match then they score and are removed from the screen. The next player then tries to find a pair until all the cards are gone. The winner is the one who correctly matched the most pairs.

In some card versions of this game a successful player is allowed to turn over another two cards. However, the game plays better in this 12-card version if one player cannot clear the decks so quickly.

Since the game requires a pack of 'micro cards' together with the shuffle routines, these can be merged from the earlier listing. Once again you will have a very comprehensive program which makes full use of the memory, without having to type in more than is necessary.

Here is a money raising tip. Those readers who are involved in raising cash for charities or school funds may like to consider how the micro can help. *Concentration* could be used as a fund raiser, just as

many other programs featured in the book can. Simply set up your computer and charge visitors for the privilege of pressing the buttons!

```
1 REM << CONCENTRATION    ANDY/JIM >>>
2 REM
10 CLEAR
20 DIM pac$(5,13),cc(3),sc(2)
30 DIM pa(52),av(52)
40 DIM cx(12),cy(12)
50 DIM ca(12),bc(12)
60 DIM plc(1,2),ac(12)
70 SYMBOL AFTER 223
80 GOSUB 17000
90 CLS:GOSUB 10000
100 INK 0,0:BORDER 0
110 sc(1)=0:sc(2)=0
120 GOSUB 10000
130 GOSUB 10500
135 PAPER 0:PEN 3
140 GOSUB 11500
150 GOSUB 12000
160 FOR a=1 TO 12:ac(a)=1:NEXT a
170 WINDOW #3,18,40,25,25
180 p=1
190 GOSUB 14500
200 GOSUB 13500
210 GOTO 14000
220 IF p=1 THEN p=2 ELSE p=1
230 GOTO 190
11295 REM >>>>>>>> ERASE CARD <<<<<<<<<
11300 LOCATE cx-1,cy
11305 FOR y=1 TO 7
11310 LOCATE cx-1,cy+y-1:PRINT "          "
11315 NEXT y
11320 RETURN
11490 REM >>>>>> GENERATE CARDS <<<<<<<
11500 RESTORE 11570:FOR a=1 TO 12
11501 ca(a)=1:NEXT a
11504 FOR a=1 TO 12 STEP 2
11505 READ c1,c2
11510 c=(RND(1)*11)+1
11515 IF ca(c)=0 THEN 11510
11520 bc(c)=c1:ca(c)=0
11525 c=(RND(1)*11)+1
11530 IF ca(c)=0 THEN 11525
```

```
11535 bc(c)=c2:ca(c)=0
11540 NEXT a
11545 FOR a=1 TO 12
11550 READ cx(a),cy(a)
11555 NEXT a
11560 RETURN
11570 DATA 1,14,2,15,3,16,4,17,5,18,6,19
11580 DATA 2,1,10,1,18,1,26,1
11585 DATA 2,9,10,9,18,9,26,9
11590 DATA 2,17,10,17,18,17,26,17
11990 REM >>>>>>> DRAW SCREEN <<<<<<<<
12000 MODE 1:INK 1,26:INK 2,6:INK 3,14
12004 FOR a=1 TO 12
12005 cx=cx(a):cy=cy(a)
12010 GOSUB 11200
12011 LOCATE cx(a)+1,cy(a)+7:PRINT a
12015 NEXT a
12020 LOCATE 33,3:PRINT"Score 1"
12025 LOCATE 33,6:PRINT "Score 2"
12030 LOCATE 32,15:PRINT "Player 1"
12035 LOCATE 32,17:PRINT "Player 2"
12040 RETURN
12990 REM >>>>>>>> DECODE CARD <<<<<<<<
13000 su=plc(q,w)/13:car=plc(q,w) MOD 13
:IF car=0 THEN car=13
13005 IF su>0 THEN suit=1
13010 IF su>1 THEN suit=2
13015 IF su>2 THEN suit=3
13020 IF su>3 THEN suit=4
13025 RETURN
13490 REM >>>>>>>> INPUT NUMBER <<<<<<<
13500 LOCATE 1,25:FOR a=1 TO 30:PRINT "
";:NEXT a
13504 LOCATE 1,25:PRINT"Enter first No."
;
13505 INPUT #3,n1
13510 IF n1<1 OR n1>12 THEN 13505
13511 IF ac(n1)=0 THEN 13505
13515 plc(1,1)=bc(n1):q=1:w=1
13520 GOSUB 13000:cx=cx(n1):cy=cy(n1)
13525 GOSUB 11000
13530 LOCATE 1,25:PRINT"Enter second No.
";
13535 INPUT #3,n2
13540 IF n2<1 OR n2>12 THEN 13535
```

```
13541 IF ac(n2)=0 THEN 13535
13545 plc(1,2)=bc(n2):q=1:w=2
13550 GOSUB 13000:cx=cx(n2):cy=cy(n2)
13555 GOSUB 11000
13560 RETURN
13990 REM >>>>>>> CHECK PAIR <<<<<<<<
14000 q=1:w=1:GOSUB 13000
14005 ca1=car:q=1:w=2:GOSUB 13000
14010 ca2=car
14015 IF ca1=ca2 THEN 15000
14020 LOCATE 1,25:FOR a=1 TO 30:PRINT" "
;:NEXT a
14025 LOCATE 1,25:PRINT "Cards do not ma
tch"
14030 GOSUB 19500
14035 cx=cx(n1):cy=cy(n1)
14040 GOSUB 11200:cx=cx(n2):cy=cy(n2)
14045 GOSUB 11200:GOTO 220
14490 REM >>>>> PRINT SCORES <<<<<<<<<
14500 PEN 1:LOCATE 33,4:PRINT sc(1)
14505 LOCATE 33,7:PRINT sc(2)
14510 IF p=1 THEN LOCATE 32,15:PAPER 1:P
EN 0:PRINT "Player 1":PEN 1:PAPER 0:LOCA
TE 32,17:PRINT "Player 2"
14515 IF p=2 THEN LOCATE 32,17:PAPER 1:P
EN 0:PRINT "Player 2":PEN 1:PAPER 0:LOCA
TE 32,15:PRINT "Player 1"
14520 RETURN
14990 REM >>>>>>>>> PAIR FOUND <<<<<<<<
15000 LOCATE 1,25:FOR a=1 TO 30:PRINT "
";:NEXT a
15005 LOCATE 1,25:PRINT "Pair found"
15006 ac(n1)=0:ac(n2)=0
15007 FOR so=500 TO 100 STEP -10:SOUND 1
,so,1:NEXT so
15010 GOSUB 19500
15015 cx=cx(n1):cy=cy(n1)
15020 GOSUB 11300:cx=cx(n2):cy=cy(n2)
15025 GOSUB 11300:sc(p)=sc(p)+5
15030 IF sc(1)+sc(2)=30 THEN 15500
15035 GOTO 220
15490 REM >>>>>>> END OF GAME <<<<<<<<<
15500 CLS:PEN 1
15505 PRINT:PRINT "Player 1 score:";sc(1
)
```

```
15510 PRINT:PRINT "Player 2 score:";sc(2
)
15515 PRINT:PRINT:LOCATE 5,10:PRINT "Pre
ss 'SPACE' for another game,"
15520 LOCATE 10,12:PRINT "or 'Q' to quit
."
15525 a$=INKEY$
15530 IF a$=" " THEN 10
15535 IF a$="Q" OR a$="q" THEN END
15540 GOTO 15525
17085 RETURN
19500 FOR t=1 TO 3000:NEXT t:RETURN
```

Now 'MERGE' the 'CARD CORE' routine. . .

16
Micro Mind

Colour Your Reasoning

Versions of this well known and widely enjoyed game can be found on all microcomputers. In writing this implementation of the game, care has been taken to produce a good looking layout that makes play enjoyable.

The object of the game is to work out what the secret 6-colour combination is, by placing coloured markers in a row.

The Amstrad will then indicate how accurate the placing was using little faces:

A happy face indicates the right colour, right place.
A sad face indicates the wrong colour, wrong place.
A straight face indicates the right colour, wrong place.

A joystick or the cursor pad is used to select the colours by moving an arrow to the colour required. The fire button or copy key then places it. The next ones are then entered in the same manner until the row is full and the result is shown.

There are ten chances to guess the correct sequence before each

game ends. The computer will then display the correct combination at the top of the screen. This will remain in view to allow analysis of where you want wrong! Pressing Y or N either starts a new game or ends the session.

The game can be played competitively by each person aiming to achieve a lower number of lines to work out the combinations.

```
10 REM <<<  MICROMIND       ANDY/JIM >>>
15 REM
20 INPUT "COLOUR (Y/N) ?",a$
22 a$=UPPER$(a$)
25 IF a$="Y" THEN gr=0 ELSE gr=1
30 REM
31 MEMORY 34990:addr=34999
32 GOSUB 29500
35 GOSUB 11500
36 cx=3:cy=25
37 PEN 1
40 FOR a=1 TO 4
45 LOCATE (a*3)-2,2:PRINT CHR$(143);CHR$
(143)
46 NEXT a
50 n=1:FOR y=4 TO 22 STEP 2:LOCATE 1,y:F
OR x=1 TO 9 STEP 3:PRINT CHR$(143);CHR$(
232);" ";:NEXT x:PRINT CHR$(143);CHR$(23
2);:n=n+1:NEXT y
55 try=1
60 GOSUB 10000
65 GOSUB 10500
70 GOSUB 12500
72 IF a$="YYYY" THEN 14000
73 a$=""
75 try=try+1:IF try=11 THEN 13000 ELSE G
OTO 65
9949 END
9990 REM >>> PICK RANDOM SEQUENCE <<<<
10000 FOR pla=1 TO 4
10005 n=INT(RND(1)*4)+2
10010 nn(pla)=n
10015 NEXT pla
10020 RETURN
10490 REM >>>>> INPUT COLOURS <<<<<<
10500 IF gr=1 THEN GOTO 10502 ELSE LOCAT
E 3,24
10501 PEN 2:PRINT CHR$(143);:PEN 3:PRINT
```

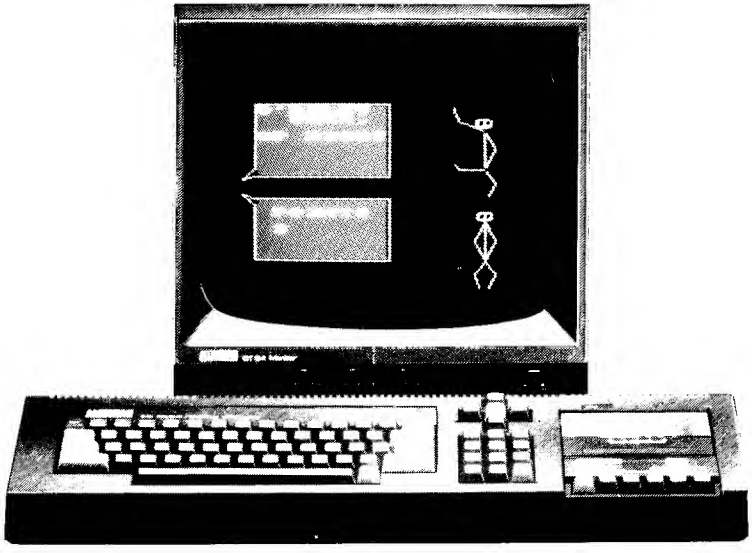
```
 CHR$(143);:PEN 4:PRINT CHR$(143);:PEN 5
:PRINT CHR$(143);:PEN 6:PRINT CHR$(143);
:PEN 7:PRINT CHR$(143):GOTO 10505
10502 LOCATE 3,24:PEN 2:PRINT CHR$(34);:
PEN 3:PRINT CHR$(35);:PEN 4:PRINT CHR$(3
6);:PEN 5:PRINT CHR$(37);:PEN 6:PRINT CH
R$(38);:PEN 7:PRINT CHR$(39)
10505 tx=1
10507 cy=25
10510 LOCATE cx,cy:PRINT "^":FOR t=0 TO
10:NEXT t
10515 LOCATE cx,cy:PRINT " "
10520 GOSUB 29000
10525 IF le=1 AND cx>3 THEN cx=cx-1
10530 IF ri=1 AND cx<8 THEN cx=cx+1
10535 IF fi=1 THEN 10600
10540 GOTO 10510
10600 pe=cx-1
10605 LOCATE tx,try*2+2:PEN pe:IF gr=1 T
HEN PRINT CHR$(32+pe) ELSE PRINT CHR$(23
3)
10610 tx=tx+3:pe(tx/3)=pe
10615 IF tx=13 THEN RETURN ELSE GOTO 105
10
11490 REM >>>>>>>> DRAW HEAD <<<<<<<<<<
11495 SYMBOL AFTER 223
11500 MODE 0:INK 1,24:INK 2,11:INK 3,18:
INK 4,24:INK 5,26:INK 6,6:INK 7,7
11501 PEN 1
11502 PEN 6:LOCATE 2,1:PRINT"MICROMIND":
PEN 1
11505 MOVE 580,50:DRAW 560,110
11510 DRAW 605,180:DRAW 615,220:DRAW 615
,250
11520 DRAW 605,290:DRAW 555,320:DRAW 485
,320
11525 DRAW 415,290:DRAW 395,250:DRAW 365
,200
11530 DRAW 375,195:DRAW 390,190:DRAW 400
,170
11535 DRAW 395,165:DRAW 400,160:DRAW 395
,155:DRAW 400,150
11540 DRAW 400,145:DRAW 395,140:DRAW 395
,130
11545 DRAW 400,125:DRAW 415,120:DRAW 425
```

```
,110
11550 DRAW 430,95:DRAW 405,50:MOVE 410,2
50:DRAW 430,245:DRAW 410,240
11555 PEN 2:LOCATE 15,10:PRINT CHR$(143)
:PEN 3:LOCATE 16,9:PRINT CHR$(143);:PEN
4:PRINT CHR$(143):PEN 5:LOCATE 18,10:PRI
NT CHR$(143):PEN 6:LOCATE 18,11:PRINT CH
R$(143)
11560 PEN 7:LOCATE 17,12:PRINT CHR$(143)
:PEN 8:LOCATE 16,13:PRINT CHR$(143):PEN
15:LOCATE 16,15:PRINT CHR$(143)
11565 SYMBOL 250,126,255,153,255,255,129
,255,126
11570 RETURN
11990 REM >>> PUT PLAYERS COLOUR UP <<<
12490 REM >>> CHECK THROUGH SEQUENCE <<
12500 a$="":tx=2:PEN 1
12505 FOR x=1 TO 4
12510 IF pe(x)=nn(x) THEN LOCATE tx,try*
2+2:PRINT CHR$(224);:a$=a$+"Y" ELSE LOCA
TE tx,try*2+2:PRINT CHR$(225);:a$=a$+"X"
12511 tx=tx+3
12515 NEXT x
12520 tx=2
12521 do=0
12525 FOR x=1 TO 4
12526 IF MID$(a$,x,1)="Y" THEN SOUND 1,1
00,10:GOTO 12550
12530 IF pe(x)=nn(1) AND LEFT$(a$,1)="X"
 THEN LOCATE tx,try*2+2:PRINT CHR$(250):
SOUND 1,300,10:do=1
12535 IF pe(x)=nn(2) AND MID$(a$,2,1)="X
" THEN LOCATE tx,try*2+2:PRINT CHR$(250)
:SOUND 1,300,10:do=1
12540 IF pe(x)=nn(3) AND MID$(a$,3,1)="X
" THEN LOCATE tx,try*2+2:PRINT CHR$(250)
:SOUND 1,300,10:do=1
12545 IF pe(x)=nn(4) AND RIGHT$(a$,1)="X
" THEN LOCATE tx,try*2+2:PRINT CHR$(250)
:SOUND 1,300,10:do=1
12546 IF do=0 THEN SOUND 1,450,10
12550 tx=tx+3:SOUND 1,0,5:NEXT x
12555 RETURN
12990 REM >>>>> LOSING SEQUENCE <<<<<
13000 FOR a=1 TO 4
```

```
13005 PEN nn(a)
13010 LOCATE (a*3)-2,2:PRINT CHR$(143);C
HR$(143)
13015 NEXT a
13025 LOCATE 13,1:PRINT "Sorry"
13026 FOR so=100 TO 500 STEP 10:SOUND 1,
so,3:NEXT so
13030 GOTO 14021
13990 REM >>>>> WINNING SEQUENCE <<<<<
14000 FOR a=1 TO 4
14005 PEN nn(a)
14010 LOCATE (a*3)-2,2:PRINT CHR$(143);C
HR$(143)
14015 NEXT a
14020 LOCATE 14,1:PRINT "Did it":LOCATE
14,3:PRINT"in ";try
14021 LOCATE 14,23:PRINT"Again":LOCATE 1
3,24:PRINT"(Y / N)"
14022 FOR a=1 TO 2
14023 FOR so=500 TO 100 STEP -50:SOUND 1
,so,5:NEXT so
14024 FOR so=100 TO 500 STEP 50:SOUND 1,
so,5:NEXT so:NEXT a
14025 a$=INKEY$
14030 IF a$="Y" OR a$="y" THEN 35
14035 IF a$="N" OR a$="n" THEN CLS:END
14040 GOTO 14025
```

Now 'MERGE' the 'INKEYS' routine. . .

17
O'Grady Says



O'Grady Says: Type This In

As far as I know this is the world's first implementation of the well known game, sometimes also known as Simon Says.

This program can be most helpful in teaching the concept of left and right, but be warned, as it stands the program means *its* left or right, not yours!

You start each game with just one point. The speech bubble will then announce the action required together with the name of the person giving the order. If O'Grady says 'do it' then you do it fast! If O'Riley says 'do it' then you do *not*! The little chap at the top follows all commands.

There are four possible actions which you may be asked to do. Up and Down are actioned using the up or down cursor or joystick as normal. Left or Right are obtained using the respective key in relation to the on-screen character. This means that pressing the left cursor will actually lift up the chap's right arm.

This swap has been incorporated to provide more of a mental challenge. It can be quite a struggle to work out which is left, which is

right and of course whether O'Grady said 'do it'. Play the game and you will see what I mean.

Speedy mental processing is important in *O'Grady Says* or lack of action may be taken as intentional. This could mean that you lose points or, if you are lucky, doing nothing could be just the right response.

```
1 REM < O'Grady says <^> ISSI/JIM/ANDY>
2 REM
100 MEMORY 34990
110 addr=34999:GOSUB 29500
500 GOSUB 20000
505 sc=0:tu=0
510 REM >>>>>>>> MAIN LOOP <<<<<<<<<<
515 og=INT(RND(1)*2)+1
516 tu=tu+1
520 po=INT(RND(1)*4)+1
521 CLS #1
522 GOSUB 21040:GOSUB 21060:GOSUB 21140:
GOSUB 21160
525 IF og=1 THEN PEN #1,4:PRINT #1,"O'GR
ADY   SAYS,"
530 IF og=2 THEN PEN #1,4:PRINT #1,"O'RI
LEY   SAYS,"
535 IF po=1 THEN PEN #1,5:PRINT #1:PRINT
 #1,"MOVE UP":GOSUB 21000:GOSUB 21020
540 IF po=2 THEN PEN #1,5:PRINT #1:PRINT
 #1,"  MOVE     DOWN":GOSUB 21060:GOSUB 2
1040
545 IF po=3 THEN PEN #1,5:PRINT #1:PRINT
 #1,"UP RIGHT":GOSUB 21000:GOSUB 21060
550 IF po=4 THEN PEN #1,5:PRINT #1:PRINT
 #1,"UP LEFT":GOSUB 21020:GOSUB 21040
555 t=0
560 GOSUB 29000
565 IF le+ri+up+do=0 THEN t=t+1:IF t=40
THEN 610 ELSE GOTO 560
570 IF ri=1 THEN GOSUB 21120:GOSUB 21140
575 IF le=1 THEN GOSUB 21100:GOSUB 21160
580 IF up=1 THEN GOSUB 21100:GOSUB 21120
585 IF do=1 THEN GOSUB 21140:GOSUB 21160
590 IF po=1 AND up=1 AND og=1 THEN 2000
595 IF po=2 AND do=1 AND og=1 THEN 2000
600 IF po=3 AND le=1 AND og=1 THEN 2000
605 IF po=4 AND ri=1 AND og=1 THEN 2000
```

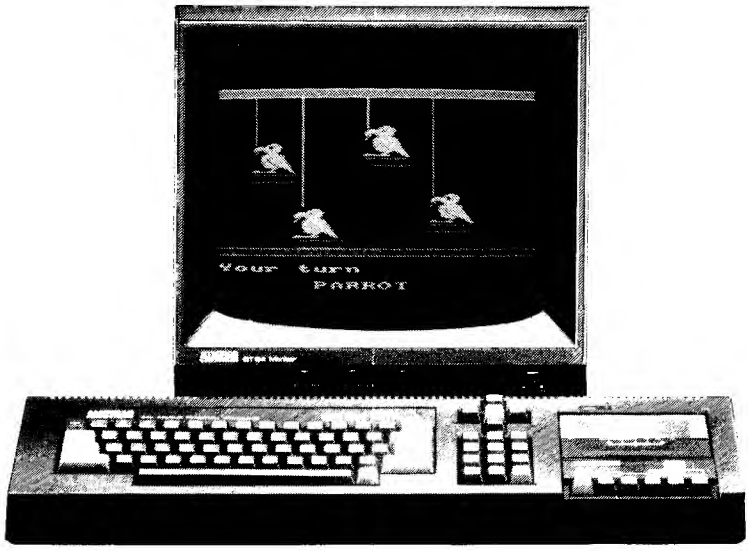
```
610 IF t=40 AND og=2 THEN CLS #1:PEN #1,
6:PRINT #1,"  WELL     DONE  ":PEN #1,4:P
RINT #1,"O'RILEY":PEN #1,6:PRINT #1,"SAI
D IT":sc=sc+1:GOSUB 2500
611 IF t=40 AND og=2 THEN FOR son=500 TO
 100 STEP -30:SOUND 1,son,10:NEXT son:GO
TO 635
615 IF og=2 AND le+ri+do+up<>0 THEN CLS
#1:PEN #1,6:PRINT #1,"WRONG!":PEN #1,4:P
RINT #1:PRINT #1,"O'RILEY":PEN #1,6:PRIN
T #1:PRINT #1,"SAID IT":sc=sc-1:GOSUB 25
00
616 IF og=2 AND le+ri+do+up<>0 THEN FOR
so=100 TO 200 STEP 10:SOUND 1,so,5:NEXT
so:GOTO 625
617 IF t=40 AND og=1 THEN CLS #1:PRINT #
1," OUT OF   TIME!":sc=sc-1:GOSUB 2500
618 IF t=40 AND og=1 THEN FOR so=100 TO
200 STEP 10:SOUND 1,so,5:NEXT so:GOTO 63
5
620 CLS #1:PRINT #1:PEN #1,5:PRINT #1,"W
RONG!"
621 FOR so=100 TO 500 STEP 50:SOUND 1,so
,10:NEXT so
625 sc=sc-1:IF sc<1 THEN 700
630 GOSUB 2500
635 FOR t=1 TO 1000:NEXT t
640 GOTO 515
690 REM >>>>> END OF GAME <<<<<<<<<<<
700 CLS #2:PEN #2,5:PRINT #2,"BAD LUCK
YOU   MADE IT THROUGH ";tu:PRINT #2," TU
RNS."
705 FOR t=1 TO 2000:NEXT t
710 CLS #1:CLS #2:PEN #2,5:PRINT #2,"POI
NTS"
715 MODE 0:GOTO 500
1990 REM >>>>>>> CORRECT <<<<<<<<<<<<
2000 CLS #1:PRINT #1,"CORRECT!"
2005 FOR so=200 TO 100 STEP -5:SOUND 1,s
o,5:NEXT so
2010 sc=sc+1:GOSUB 2500:GOTO 635
2490 REM >>>>>>> PRINT SCORE <<<<<<<<<
2500 PRINT #2,CHR$(30):PRINT #2:PRINT #2
:PEN #2,4:PRINT #2,sc
2505 RETURN
```

```
19980 REM >>>>>>>>>> screen <<<<<<<<<<
20000 RESTORE 20000
20010 FOR n=0 TO 15
20020 READ a
20030 INK n,a
20040 NEXT n
20050 DATA 0,26,3,18,24,24,24,3
20060 DATA 0,0,0,0,0,0,0,0
20100 BORDER 0:PAPER 0:PEN 1:MODE 0
20110 WINDOW #1,2,9,3,10
20120 WINDOW #2,2,9,14,20
20130 PAPER #1,2:PEN #1,5:CLS #1
20140 PAPER #2,2:PEN #2,5:CLS #2
20150 PLOT 30,370,5:DRAW 292,370
20160 DRAW 292,237:DRAW 30,237
20170 DRAW 5,230:DRAW 30,252:DRAW 30,370
20180 PLOT 30,195:DRAW 292,195
20190 DRAW 292,77:DRAW 30,77
20200 DRAW 30,180:DRAW 5,202:DRAW 30,195
20210 PRINT #2:PRINT #2," POINTS"
20220 PLOT 480,325,5:DRAW 480,250
20230 DRAW 530,250,9:DRAW 540,260
20240 PLOT 480,325:DRAW 530,340
20250 DRAW 540,365:PLOT 480,250,11
20260 DRAW 430,250:DRAW 420,260
20270 PLOT 480,325:DRAW 430,340
20280 DRAW 420,365
20290 PLOT 480,325,8:DRAW 500,290
20300 DRAW 480,250:DRAW 500,225
20310 DRAW 490,200
20320 PLOT 480,325,10:DRAW 460,290
20330 DRAW 480,250:DRAW 460,225
20340 DRAW 470,200
20500 PLOT 480,150,5:DRAW 480,75
20510 DRAW 530,75,13:DRAW 540,85
20520 PLOT 480,150:DRAW 530,165
20530 DRAW 540,190:PLOT 480,75,15
20540 DRAW 430,75:DRAW 420,85
20550 PLOT 480,150:DRAW 430,165
20560 DRAW 420,190
20570 PLOT 480,150,12:DRAW 500,115
20580 DRAW 480,75:DRAW 500,50
20590 DRAW 490,25
20600 PLOT 480,150,14:DRAW 460,115
20610 DRAW 480,75:DRAW 460,50
```

```
20620 DRAW 470,25
20630 TAG #3:PLOT 0,0,5
20640 MOVE 466,344:PRINT #3,CHR$(224);
20650 MOVE 466,169:PRINT #3,CHR$(224);
20660 PLOT 0,0,0
20670 GOSUB 21040:GOSUB 21060
20680 GOSUB 21140:GOSUB 21160
20750 RETURN
21000 INK 11,24:INK 10,0
21010 RETURN
21020 INK 9,24:INK 8,0
21030 RETURN
21040 INK 10,24:INK 11,0
21050 RETURN
21060 INK 8,24:INK 9,0
21070 RETURN
21100 INK 15,24:INK 14,0
21110 RETURN
21120 INK 13,24:INK 12,0
21130 RETURN
21140 INK 14,24:INK 15,0
21150 RETURN
21160 INK 12,24:INK 13,0
21170 RETURN
```

Now 'MERGE' the 'INKEYS' routine. . .

18
Parrot



Who's a Pretty Program?

Here are four pretty parrots, each a different colour and each holding a 'string' in his mouth.

When the program runs the parrots will commence pulling the strings in sequence. The first set will consist of only one pull. Then it will be your turn to copy the sequence.

The perches of the birds have been arranged to correspond to the arrangement of the cursor keys. Thus by pressing 'up' the top bird will pull a string, 'down' will activate the bottom bird and so on. As usual a joystick may be used instead of the cursor pad.

The computer produces a sequence which starts off simply but gets very long. During testing of this program no one was able to remember more than a sequence of 25. Amnesia usually sets in at around 12. As it stands the longest sequence that the parrots will produce is 50 long. If you remember that length then you must have some sort of photographic memory – or perhaps you are an android!

This simple game can be great fun at parties where all sorts of forfeits or penalties may be devised for the losers.

```
2 REM <<< PARROT          ANDY / JIM >>>>>
3 REM
5 addr=34999:GOSUB 29500
10 SYMBOL AFTER 223:MODE 0
12 DIM no(50)
13 INK 0,0:BORDER 0:INK 12,12
15 GOSUB 10000:GOSUB 10500
25 sc=0:hi=0
30 GOSUB 12500
35 FOR a=1 TO 4
36 IF a=1 THEN PEN 7 ELSE IF a=2 THEN PE
N 11 ELSE IF a=3 THEN PEN 14 ELSE PEN 13

40 GOSUB 11000:NEXT a
55 po=0:CALL &BB03
58 REM >>>>>>>> MAIN LOOP <<<<<<<<
60 n=INT(RND(1)*3)+1
62 po=po+1
65 no(po)=n
66 GOSUB 13500:LOCATE 1,22:PRINT "My tur
n"
70 FOR b=1 TO po
75 GOSUB 13000
80 NEXT b
82 CALL &BB03
85 FOR t=1 TO 1000:NEXT t
87 b=1
88 GOSUB 13500:PEN 1:LOCATE 1,22:PRINT "
Your turn"
90 GOSUB 29000
95 IF le=1 THEN a=4:SOUND 1,800,10:GOSUB
 12000:PEN 13:GOSUB 11500:FOR t=1 TO 500
:NEXT t
100 IF ri=1 THEN a=2:SOUND 1,400,10:GOSU
B 12000:PEN 11:GOSUB 11500:FOR t=1 TO 50
0:NEXT t
105 IF up=1 THEN a=1:SOUND 1,200,10:GOSU
B 12000:PEN 7:GOSUB 11500:FOR t=1 TO 500
:NEXT t
110 IF do=1 THEN a=3:SOUND 1,600,10:GOSU
B 12000:PEN 14:GOSUB 11500:FOR t=1 TO 50
0:NEXT t
115 IF le+ri+up+do<1 THEN 90
120 GOSUB 12000
125 GOSUB 11000
```

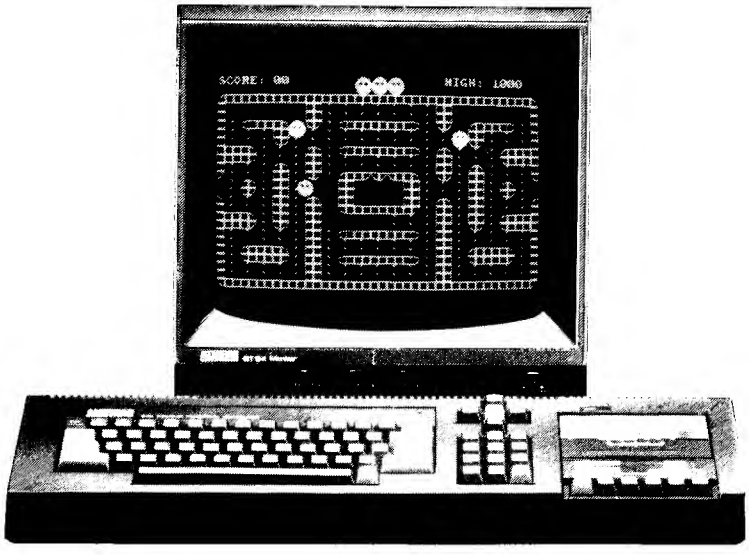
```
130 IF no(b)=a THEN 131 ELSE 135
131 IF b=po THEN GOSUB 13500:LOCATE 1,22
:PEN 1:PRINT "Well done,";po" so far."
132 IF b=po THEN FOR t=1 TO 1500:NEXT t:
GOSUB 13500:GOTO 60
133 b=b+1:GOTO 90
135 GOSUB 13500:PEN 1
140 LOCATE 1,22:PRINT "Wrong sequence"
145 LOCATE 1,23:PRINT "After";po-1;" mov
ements."
150 FOR t=1 TO 3000:NEXT t
155 GOSUB 13500
160 LOCATE 1,22:PRINT "Sequence was"
165 FOR t=1 TO 1000:NEXT t
170 FOR b=1 TO po
175 GOSUB 13000
180 NEXT b
185 GOSUB 13500
190 LOCATE 1,23:PRINT "Try again."
195 FOR t=1 TO 2000:NEXT t
200 GOSUB 13500
205 GOTO 55
9990 REM >>>>>> CHARACTERS <<<<<<<<<<
10000 SYMBOL 224,16,16,16,16,0,14,31,63
10005 SYMBOL 225,0,0,28,126,127,207,207,
207
10010 SYMBOL 226,0,0,0,0,0,128,128
10015 SYMBOL 227,63,49,40,8,16,33,35,3
10020 SYMBOL 228,255,255,254,124,126,255
,255,255
10025 SYMBOL 229,0,0,0,0,0,128,192,192
10030 SYMBOL 230,3,3,3,3,3,1,1,1
10035 SYMBOL 231,223,191,191,223,223,239
,247,251
10040 SYMBOL 232,192,192,224,224,224,240
,240,248
10045 SYMBOL 233,1,0,0,0,56,127,0,0
10050 SYMBOL 234,253,126,12,24,48,252,0,
0
10055 SYMBOL 235,248,120,60,28,14,6,3,0
10060 SYMBOL 236,63,30,0,0,7,15,0,0
10065 SYMBOL 237,248,0,0,0,0,0,0,0
10070 SYMBOL 238,7,31,63,127,99,99,127,6
3
10075 SYMBOL 239,144,225,247,251,139,115
```

```
,251,255
10080 SYMBOL 240,0,254,255,255,255,255,2
55,255
10085 SYMBOL 241,3,14,252,252,248,240,22
4,128
10090 SYMBOL 242,16,16,16,16,16,16,16,16
10095 SYMBOL 243,255,170,85,170,85,170,8
5,255
10490 REM >>> SET UP VARIABLES <<<<<<<<
10500 px(1)=10:px(2)=14:px(3)=6:px(4)=3
10505 py(1)=5:py(2)=13:py(3)=15:py(4)=7
10510 RETURN
10990 REM >>> PRINT UPRIGHT PARROT <<<<
11000 LOCATE px(a),py(a)
11010 PRINT CHR$(224);CHR$(225);CHR$(226
)
11015 LOCATE px(a),py(a)+1
11020 PRINT CHR$(227);CHR$(228);CHR$(229
)
11025 LOCATE px(a),py(a)+2
11030 PRINT CHR$(230);CHR$(231);CHR$(232
)
11035 LOCATE px(a),py(a)+3
11040 PRINT CHR$(233);CHR$(234);CHR$(235
)
11045 RETURN
11490 REM >>> PRINT BENT DOWN PARROT <<
11500 LOCATE px(a)-1,py(a)+2
11510 PRINT CHR$(238);CHR$(239);CHR$(240
);CHR$(241)
11515 LOCATE px(a)-1,py(a)+3
11520 PRINT CHR$(236);CHR$(234);CHR$(237
)
11521 LOCATE px(a),py(a)+1:PRINT CHR$(24
2)
11522 LOCATE px(a),py(a):PRINT CHR$(242)

11525 RETURN
11990 REM >>>>> BLANK PARROT <<<<<
12000 LOCATE px(a),py(a):PRINT "   "
12005 LOCATE px(a),py(a)+1:PRINT "   "
12010 LOCATE px(a)-1,py(a)+2:PRINT " "
12015 LOCATE px(a)-1,py(a)+3:PRINT " ":L
OCATE px(a)+2,py(a)+3:PRINT " "
12020 RETURN
```

```
12490 REM >>>>>>> DRAW SCREEN <<<<<<<<
12500 FOR y=1 TO 4:PEN 7:LOCATE 10,y:PRI
NT CHR$(242):NEXT y
12505 FOR y=1 TO 12:PEN 11:LOCATE 14,y:P
RINT CHR$(242):NEXT y
12510 FOR y=1 TO 14:PEN 14:LOCATE 6,y:PR
INT CHR$(242):NEXT y
12515 FOR y=1 TO 6:PEN 13:LOCATE 3,y:PRI
NT CHR$(242):NEXT y
12520 FOR x=1 TO 20:LOCATE x,1:PEN 15:PR
INT CHR$(243):LOCATE x,20:PEN 3:PRINT CH
R$(243):NEXT x
12530 LOCATE 10,9:PEN 12:PRINT CHR$(243)
;CHR$(243);CHR$(243)
12535 LOCATE 14,17:PRINT CHR$(243);CHR$(
243);CHR$(243)
12540 LOCATE 6,19:PRINT CHR$(243);CHR$(2
43);CHR$(243)
12545 LOCATE 3,11:PRINT CHR$(243);CHR$(2
43);CHR$(243)
12550 INK 15,24:INK 13,18:INK 14,14
12560 PEN 13:LOCATE 7,24:PRINT "PARROT";
12600 RETURN
12990 REM >>>>>>>> MAKE SOUND <<<<<<<<
13000 IF no(b)=1 THEN a=1:GOSUB 12000:PE
N 7:GOSUB 11500:SOUND 1,200,10:FOR t=1 T
O 500:NEXT t
13005 IF no(b)=2 THEN a=2:GOSUB 12000:PE
N 11:GOSUB 11500:SOUND 1,400,10:FOR t=1
TO 500:NEXT t
13010 IF no(b)=3 THEN a=3:GOSUB 12000:PE
N 14:GOSUB 11500:SOUND 1,600,10:FOR t=1
TO 500:NEXT t
13015 IF no(b)=4 THEN a=4:GOSUB 12000:PE
N 13:GOSUB 11500:SOUND 1,800,10:FOR t=1
TO 500:NEXT t
13020 GOSUB 12000:GOSUB 11000
13025 RETURN
13450 REM
13500 LOCATE 1,22:FOR q=1 TO 20:PRINT "
";:NEXT q
13505 LOCATE 1,23:FOR q=1 TO 20:PRINT "
";:NEXT q
13510 RETURN
```

Now 'MERGE' the 'INKEYS' routine. . .

19
Pick Man



An Arcade Gobble

This is the first of three arcade type games included in this book. It is a variation on a well known theme in which you guide your character around a maze eating up the dots as you go.

Life is never easy in the arcade world and so there are a couple of nasties out to eat up the dots before you do.

To make matters even worse they will kill you if they get you! The object of this variant is to pick up as many dots as possible before all the dots are gone, without being killed. Once all the dots are gone the next screen will be displayed.

Controls are, as usual, joystick or cursor pad, which change the direction of the Pick Man.

In the hands of a skilled and patient player, incredible high scores can be achieved. The 'High Score' subroutine can provide a *Pick Man* Hall Of Fame.

```
1 REM <<<<<<< pick man <^> ISSI >>>>>>>
2 REM
500 SYMBOL 240,3,15,63,63,123,113,241,25
1
510 SYMBOL 241,192,240,252,252,222,142,1
43,223
520 SYMBOL 242,255,255,127,127,63,63,15,
3
530 SYMBOL 243,255,255,254,254,252,252,2
40,192
550 SYMBOL 244,3,15,63,63,123,113,241,25
1
560 SYMBOL 245,192,240,252,252,222,142,1
43,223
570 SYMBOL 246,255,255,127,127,63,63,15,
3
580 SYMBOL 247,255,255,254,254,252,252,2
40,192
600 SYMBOL 248,3,29,49,33,65,65,129,255

610 SYMBOL 249,255,129,65,65,33,49,29,3

620 SYMBOL 250,192,176,140,132,130,130,1
29,255
630 SYMBOL 251,255,129,130,130,132,140,1
76,192
640 SYMBOL 252,255,129,129,129,129,129,1
29,255
650 SYMBOL 253,3,3,0,0,0,0,0,0
660 SYMBOL 254,15,112,64,128,128,64,112,
15
670 SYMBOL 255,240,14,2,1,1,2,14,240
750 hi=1000
980 REM >>>>>>>>>> main loop <<<<<<<<<<
1000 GOSUB 21000
1040 sc=0:count=8:li=3
1060 GOSUB 20000:t$=CHR$(240)+CHR$(241):
b$=CHR$(242)+CHR$(243)
1065 t1$=CHR$(244)+CHR$(245):b1$=CHR$(24
6)+CHR$(247)
1070 PEN 2
1080 LOCATE 18,1:PRINT t$;t$;t$
1090 LOCATE 18,2:PRINT b$;b$;b$
1500 x=20:y=17
1510 x1=2:y1=4:x2=38:y2=4
```

```
2000 GOSUB 22000
2020 GOSUB 25500
2040 GOSUB 22000
2050 IF (x1=x AND y1=y) OR (x2=x AND y2=
y) THEN dead=1
2070 IF count>258 THEN GOTO 24000
2080 IF dead=1 THEN GOTO 24500
3000 GOTO 2000
19980 REM >>>>>>>>>> screen <<<<<<<<<<
20000 INK 0,0:INK 1,18,19:INK 2,6,16:INK
 3,20,2
20005 SPEED INK 6,6
20010 PAPER 0:PEN 1:MODE 1:BORDER 0
20020 PRINT "SCORE: 00
 HIGH:";hi:PRINT
20030 w$=CHR$(252)+CHR$(252):x$=CHR$(248
)+CHR$(250):v$=CHR$(253)+CHR$(32)
20035 y$=CHR$(249)+CHR$(251):z$=CHR$(254
)+STRING$(8,252)+CHR$(255)
20040 r$=v$+v$+v$:s$=r$+v$+v$:a$=CHR$(24
8)+STRING$(38,252)+CHR$(250)
20050 PRINT a$;
20060 a$=CHR$(252)+s$+w$+s$+v$+v$+w$+s$+
CHR$(252)
20070 PRINT a$;:q$=a$
20080 PRINT a$;
20090 a$=CHR$(252)+v$+CHR$(254)+w$+w$+CH
R$(250)+v$+y$+v$+z$+v$+y$+v$+CHR$(248)+w
$+w$+CHR$(255)+v$+CHR$(252)
20100 PRINT a$;
20110 a$=CHR$(252)+r$+w$+s$+s$+v$+w$+r$+
CHR$(252)
20120 PRINT a$;
20130 MID$(a$,8,2)=y$:MID$(a$,32,2)=y$
20140 PRINT a$;
20150 a$=w$+w$+CHR$(250)+r$+x$+v$+z$+v$+
x$+r$+CHR$(248)+w$+w$
20160 PRINT a$;
20170 a$=w$+w$+CHR$(251)+r$+w$+s$+v$+v$+
w$+r$+CHR$(249)+w$+w$
20180 PRINT a$;
20190 a$=CHR$(252)+r$+x$+v$+w$+s$+v$+v$+
w$+v$+x$+r$+CHR$(252)
20200 PRINT a$;
20210 a$=CHR$(251)+r$+w$+v$+y$+v$+CHR$(2
```

```
48)+w$+CHR$(251)+y$
20220 a$=a$+CHR$(249)+w$+CHR$(250)+v$+y$
+v$+w$+r$+CHR$(249)
20230 PRINT a$;
20240 a$=CHR$(32)+v$+x$+v$+w$+r$+w$+STRI
NG$(6,32)+w$+r$+w$+v$+x$+v$
20250 PRINT a$;
20260 MID$(a$,4,2)=y$:MID$(a$,36,2)=y$
20270 PRINT a$;
20280 a$=CHR$(250)+r$+w$+v$+x$+v$+w$+STR
ING$(6,32)+w$+v$+x$+v$+w$+r$+CHR$(248)
20290 PRINT a$;
20300 a$=CHR$(252)+MID$(a$,2,10)+w$+v$+C
HR$(249)+STRING$(8,252)+CHR$(251)+v$+w$+
MID$(a$,30,10)+CHR$(252)
20310 PRINT a$;
20320 a$=w$+w$+CHR$(250)+v$+w$+v$+w$+s$+
v$+v$+w$+v$+w$+v$+CHR$(248)+w$+w$
20330 PRINT a$;
20340 MID$(a$,5,1)=CHR$(251):MID$(a$,36,
1)=CHR$(249):MID$(a$,8,2)=y$:MID$(a$,32,
2)=y$
20350 PRINT a$;
20360 a$=CHR$(252)+s$+w$+v$+z$+v$+w$+s$+
CHR$(252)
20370 PRINT a$;
20380 a$=q$
20390 PRINT a$;
20400 b$=CHR$(248)+w$+w$+CHR$(250)
20410 MID$(a$,4,6)=b$:MID$(a$,32,6)=b$
20420 PRINT a$;
20430 b$=CHR$(249)+w$+w$+CHR$(251)
20440 MID$(a$,4,6)=b$:MID$(a$,32,6)=b$
20450 MID$(a$,16,10)=z$
20460 PRINT a$;
20470 PRINT q$;:PRINT q$;
20480 a$=CHR$(249)+STRING$(38,252)+CHR$(
251)
20490 PRINT a$;
20500 RETURN
20980 REM >>>>>>>>> initialize <<<<<<<<<
21000 MEMORY 34998
21010 addr=34999:GOSUB 29500
21020 loca=35029:GOSUB 30500
21250 RETURN
```

```
21980 REM >>>>>>>>>> movement <<<<<<<<<
22000 GOSUB 29000
22010 IF le=1 THEN GOSUB 23000
22020 IF ri=1 THEN GOSUB 23200
22030 IF up=1 THEN GOSUB 23400
22040 IF do=1 THEN GOSUB 23600
22190 GOSUB 26000
22200 RETURN
22980 REM >>>>>>>>>> checks <<<<<<<<<<
23000 LOCATE x-1,y:GOSUB 32000
23010 IF ch<>32 THEN RETURN
23050 LOCATE x-1,y+1:GOSUB 32000
23060 IF ch<>32 THEN RETURN
23100 GOSUB 23800:x=x-1:IF x=1 THEN x=38
23150 RETURN
23200 LOCATE x+2,y:GOSUB 32000
23210 IF ch<>32 THEN RETURN
23250 LOCATE x+2,y+1:GOSUB 32000
23260 IF ch<>32 THEN RETURN
23300 GOSUB 23800:x=x+1:IF x=39 THEN x=2
23350 RETURN
23400 LOCATE x,y-1:GOSUB 32000
23410 IF ch<>32 THEN RETURN
23450 LOCATE x+1,y-1:GOSUB 32000
23460 IF ch<>32 THEN RETURN
23500 GOSUB 23800:y=y-1
23550 RETURN
23600 LOCATE x,y+2:GOSUB 32000
23610 IF ch<>32 THEN RETURN
23650 LOCATE x+1,y+2:GOSUB 32000
23660 IF ch<>32 THEN RETURN
23700 GOSUB 23800:y=y+1
23750 RETURN
23800 LOCATE x,y:PRINT "  "
23810 LOCATE x,y+1:PRINT "  "
23820 RETURN
23850 LOCATE x1,y1:PRINT "  ":LOCATE x1,
y1+1:PRINT "  "
23860 LOCATE x2,y2:PRINT "  ":LOCATE x2,
y2+1:PRINT "  "
23890 RETURN
23980 REM >>>>>>>>>> clear <<<<<<<<<<
24000 sc=sc+1000:LOCATE 6,1:PRINT sc
24020 RESTORE 40100:tune=12:GOSUB 31000
24080 count=0
```

```
24090 GOTO 1060
24480 REM >>>>>>>>>>> dead <<<<<<<<<<
24500 c$=t$:d$=b$:dead=0
24510 LOCATE x,y:PRINT "  ":LOCATE x,y+1
:PRINT "  "
24530 RESTORE 40000:tune=8:GOSUB 31000
24540 GOSUB 23850:PEN 2
24550 li=li-1:IF li=0 THEN GOTO 35000
24560 LOCATE 18,1:PRINT c$:LOCATE 18,2:P
RINT d$
24570 IF li=1 THEN c$="  ":d$="  "
24580 LOCATE 20,1:PRINT c$:LOCATE 20,2:P
RINT d$
24590 IF li=2 THEN c$="  ":d$="  "
24600 LOCATE 22,1:PRINT c$:LOCATE 22,2:P
RINT d$
24610 GOTO 1500
24980 REM >>>>>>> move baddies <<<<<<<
25000 LOCATE x1-1,y1:GOSUB 30000:IF ch<>
253 AND ch>247 THEN RETURN
25005 IF ch=253 THEN count=count+1
25010 LOCATE x1-1,y1+1:GOSUB 30000:IF ch
<>253 AND ch>247 THEN RETURN
25015 IF ch=253 THEN count=count+1
25020 GOSUB 23850:x1=x1-1:GOSUB 26040:RE
TURN
25050 LOCATE x2-1,y2:GOSUB 30000:IF ch<>
253 AND ch>247 THEN RETURN
25055 IF ch=253 THEN count=count+1
25060 LOCATE x2-1,y2+1:GOSUB 30000:IF ch
<>253 AND ch>247 THEN RETURN
25065 IF ch=253 THEN count=count+1
25070 GOSUB 23850:x2=x2-1:GOSUB 26040:RE
TURN
25100 LOCATE x1+2,y1:GOSUB 30000:IF ch<>
253 AND ch>247 THEN RETURN
25105 IF ch=253 THEN count=count+1
25110 LOCATE x1+2,y1+1:GOSUB 30000:IF ch
<>253 AND ch>247 THEN RETURN
25115 IF ch=253 THEN count=count+1
25120 GOSUB 23850:x1=x1+1:GOSUB 26040:RE
TURN
25150 LOCATE x2+2,y2:GOSUB 30000:IF ch<>
253 AND ch>247 THEN RETURN
25155 IF ch=253 THEN count=count+1
```

```
25160 LOCATE x2+2,y2+1:GOSUB 30000:IF ch
<>253 AND ch>247 THEN RETURN
25165 IF ch=253 THEN count=count+1
25170 GOSUB 23850:x2=x2+1:GOSUB 26040:RE
TURN
25200 LOCATE x1,y1-1:GOSUB 30000:IF ch<>
253 AND ch>247 THEN RETURN
25205 IF ch=253 THEN count=count+1
25210 LOCATE x1+1,y1-1:GOSUB 30000:IF ch
<>253 AND ch>247 THEN RETURN
25215 IF ch=253 THEN count=count+1
25220 GOSUB 23850:y1=y1-1:GOSUB 26040:RE
TURN
25250 LOCATE x2,y2-1:GOSUB 30000:IF ch<>
253 AND ch>247 THEN RETURN
25255 IF ch=253 THEN count=count+1
25260 LOCATE x2+1,y2-1:GOSUB 30000:IF ch
<>253 AND ch>247 THEN RETURN
25265 IF ch=253 THEN count=count+1
25270 GOSUB 23850:y2=y2-1:GOSUB 26040:RE
TURN
25300 LOCATE x1,y1+2:GOSUB 30000:IF ch<>
253 AND ch>247 THEN RETURN
25305 IF ch=253 THEN count=count+1
25310 LOCATE x1+1,y1+2:GOSUB 30000:IF ch
<>253 AND ch>247 THEN RETURN
25315 IF ch=253 THEN count=count+1
25320 GOSUB 23850:y1=y1+1:GOSUB 26040:RE
TURN
25350 LOCATE x2,y2+2:GOSUB 30000:IF ch<>
253 AND ch>247 THEN RETURN
25355 IF ch=253 THEN count=count+1
25360 LOCATE x2+1,y2+2:GOSUB 30000:IF ch
<>253 AND ch>247 THEN RETURN
25365 IF ch=253 THEN count=count+1
25370 GOSUB 23850:y2=y2+1:GOSUB 26040:RE
TURN
25400 RETURN
25500 IF RND(1)>0.7 THEN GOTO 25550
25510 IF x>x1 THEN GOSUB 25100
25520 IF x<x1 THEN GOSUB 25000
25530 IF y>y1 THEN GOSUB 25300
25540 IF y<y1 THEN GOSUB 25200
25550 IF RND(1)>0.6 THEN RETURN
25560 IF x>x2 THEN GOSUB 25150
```

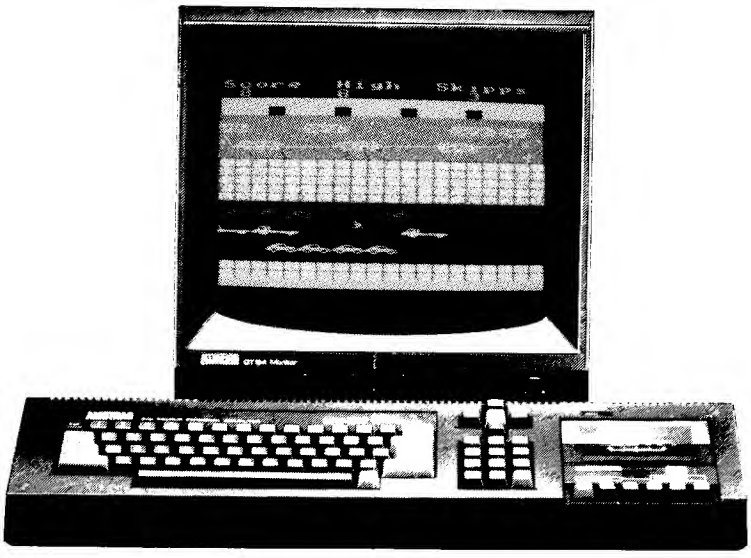
```
25570 IF x<x2 THEN GOSUB 25050
25580 IF y>y2 THEN GOSUB 25350
25590 IF y<y2 THEN GOSUB 25250
25600 RETURN
25980 REM >>>>>>>>> print man <<<<<<<<<
26000 PEN 2
26010 LOCATE x,y:PRINT t$
26020 LOCATE x,y+1:PRINT b$
26030 RETURN
26040 PEN 3
26050 LOCATE x1,y1:PRINT t1$
26060 LOCATE x1,y1+1:PRINT b1$
26070 LOCATE x2,y2:PRINT t1$
26080 LOCATE x2,y2+1:PRINT b1$
30980 REM >>>>>>>>> play tune <<<<<<<<<
31000 FOR n=1 TO tune
31010 READ note,dura
31020 SOUND 2,note,dura,15
31030 NEXT n
31040 RETURN
31980 REM >>>>>>>>> detection <<<<<<<<<
32000 GOSUB 30000
32010 PEN 1
32020 IF ch=253 THEN ch=32:sc=sc+10:SOUN
D 2,120,1,15:count=count+1:LOCATE 7,1:PR
INT sc
32030 IF ch=32 THEN PRINT " ":RETURN
32040 IF ch>247 THEN RETURN
32050 dead=1
32090 RETURN
34980 REM >>>>>>>>>>> end game <<<<<<<<
35000 CLS
35010 PEN 1
35020 LOCATE 8,3:PRINT "YOU HAVE BEEN GO
BBLED !"
35030 LOCATE 8,21:PRINT "Press 'Y' to pl
ay Again"
35040 IF sc>hi THEN hi=sc:LOCATE 10,9:PR
INT "A new high gobble !"
35150 a$=INKEY$:a$=UPPER$(a$)
35160 IF a$<>"Y" THEN GOTO 35150
35170 GOTO 1000
39990 REM * * * dead tune * * *
40000 DATA 284,20,284,20,284,20,358,75,3
19,20,319,20,319,20,379,75
```

```
40090 REM * * * screen tune * * *
40100 DATA 319,20,253,20,213,50,253,20,2
13,20,253,50
40110 DATA 319,20,253,20,213,50,253,20,3
19,20,253,50
```

Now 'MERGE' the 'INKEYS' and the

'CHAR CHECK' routines. . .

20
Skippy



One Jump Ahead

'Once upon a time in Australia, which is a long way from Britain, there lived a little kangaroo. This little chap was called Skippy and he was very happy for many years.

'Skippy had his home by the banks of a fast flowing river on which logs floated. He used to enjoy sitting in his little hut looking out across the beautiful countryside, but one day things changed.

'The nasty humans came, and built a car expressway right next to the river. This meant that every day when Skippy and his friends returned from foraging for food, they had to cross the busy road, jump onto the logs to cross the river and then jump off as they passed their home. This is very dangerous and so they would like you, dear readers, to help.'

Using the joystick or keyboard guide Skippy and his friends safely between the cars, onto the logs and into their homes. Your assistance to wildlife will be rewarded with points in relation to how well you guide each kangaroo!

This game can be merged with the 'High Score' program to add those extra features to it.

```
10 REM <<< SKIPPY          ANDY / JIM >>>>
20 REM
30 SYMBOL AFTER 223
40 MEMORY 34990
60 GOSUB 850
65 gam$="Skippy"
70 REM
110 REM >>>> DEFINE OBJECTS <<<<<<<<<<
120 car$=CHR$(224)+CHR$(225)
130 lor$=CHR$(228)+CHR$(227)+CHR$(226)
140 bik$=CHR$(241)+CHR$(242)
150 lo$=CHR$(229)+CHR$(230)+CHR$(231)
155 gap$=CHR$(245)+CHR$(245)+CHR$(245)
160 g$=STRING$(2," ")
165 h$=STRING$(3," ")
170 c1$=car$+car$+g$+g$+g$+g$+car$+g$+g$
+car$+car$+car$+g$+car$+g$+car$+car$+g$+
g$+g$+car$+g$+g$+car$+g$
175 c1$=c1$+car$+car$+g$+g$+g$+car$+car$
+car$+car$+g$+g$+g$+g$+car$+g$+car$+g$+c
ar$+car$+g$+g$+g$+car$+g$+g$+car$+car$
180 c2$=lor$+h$+h$+h$+h$+lor$+h$+lor$+lo
r$+h$+h$+lor$+h$+h$+h$+lor$+h$+h$+h$+h$+
lor$+h$+h$+h$+lor$+lor$+lor$+h$+h$+h$+h$
+"  "+lor$+h$
190 c3$=bik$+bik$+h$+h$+g$+bik$+bik$+bik
$+bik$+bik$+h$+h$+g$+bik$+bik$+bik$+bik$
+bik$+h$+h$+" "+bik$+g$+bik$+h$+" "+bik$
+bik$+bik$+h$+" "+bik$+h$+h$+h$+g$+bik$+
bik$+h$+h$+g$+bik$+bik$+bik$+bik$
200 l1$=lo$+gap$+gap$+gap$+lo$+gap$+lo$+
gap$+lo$+gap$+lo$+gap$+gap$+gap$+gap$+lo
$+gap$+gap$+lo$+lo$+gap$+gap$+gap$+lo$+l
o$+lo$+CHR$(245)+CHR$(245)+lo$+gap$+gap$
+gap$+lo$+gap$+lo$
210 l2$=lo$+lo$+gap$+gap$+lo$+gap$+lo$+g
ap$+gap$+lo$+lo$+gap$+lo$+gap$+gap$+lo$+
lo$+gap$+lo$+lo$+gap$+gap$+lo$+gap$+lo$+
gap$+gap$+lo$+lo$+gap$+gap$+lo$+gap$+lo$
215 FOR a=1 TO 100:riv$=riv$+CHR$(245):N
EXT a
220 FOR a=1 TO 20:ban1$=ban1$+CHR$(243):
```

```
NEXT a
230 FOR a=1 TO 20 STEP 4:ban1$=ban1$+CHR
$(243)+CHR$(243)+CHR$(243)+" ":NEXT a:ba
n1$=LEFT$(ban1$,39):ban1$=ban1$+CHR$(243
)
240 FOR a=1 TO 100:ban2$=ban2$+CHR$(244)
:NEXT a
250 FOR a=1 TO 40:ban3$=ban3$+CHR$(244):
NEXT a
255 FOR a=1 TO 20:ban4$=ban4$+CHR$(243):
NEXT a
260 hi=0
300 ka=3:sc=0
310 GOSUB 710
315 REM >>>>> SET UP VARIABLES <<<<<
320 c1=-1:c2=1:c3=-1:l1=1:l2=-1
330 cc1=80:cc2=1:cc3=80:ll1=70:ll2=10
340 k=232:k1=232:k2=233
350 kx=10:ky=22
360 ox=10:oy=22
362 kab=0
365 REM >>>>>>> MAIN GAME LOOP <<<<<<<
370 GOSUB 570
373 ret=1:GOTO 740
375 ox=kx:oy=ky
376 GOSUB 40000
400 cc1=cc1+c1:cc2=cc2+c2:cc3=cc3+c3
407 ll1=ll1+l1:ll2=ll2+l2
420 IF cc1=1 THEN cc1=81
430 IF cc2=81 THEN cc2=1
440 IF cc3=1 THEN cc3=81
450 IF ll1=81 THEN ll1=1
460 IF ll2=1 THEN ll2=81
461 IF ky=8 THEN kdx=-1
462 IF ky=6 THEN kdx=1
463 IF kx+kdx=0 OR kx+kdx=21 THEN 505
464 ret=2:GOTO 740
510 ox=kx:oy=ky:GOTO 370
560 REM >>>>>>> PRINT OBSTRUCTIONS <<<<
570 PRINT CHR$(30):PRINT:INK 2,20
580 PRINT:PRINT:PRINT:PEN 10:PAPER 0:PRI
NT MID$(L2$,LL2,19)
590 PRINT:PRINT MID$(L1$,LL1,19):PRINT:P
RINT:PRINT:PRINT
600 PRINT:PRINT:PRINT:PAPER 0:PEN 8:PRIN
```

```
T MID$(C3$,CC3,19)
610 PRINT:PEN 2:PRINT MID$(C2$,CC2,19)
620 PRINT:PEN 7:PRINT MID$(C1$,CC1,19)
630 RETURN
650 REM >>>>>>> PRINT KANGAROO <<<<<<<
690 PEN 12:PAPER 0:LOCATE kx,ky:PRINT CH
R$(k):RETURN
692 REM >>>>>>> ERASE KANGAROO <<<<<<<
693 IF oy<5 THEN cc=243:PAPER 0:PEN 15
694 IF oy=8 OR oy=6 THEN cc=0:GOTO 700
695 IF oy>4 AND oy<11 THEN cc=207:PAPER
10:PEN 14
696 IF oy>9 AND oy<15 THEN cc=244:PAPER
0:PEN 14
697 IF oy>14 AND oy<22 THEN cc=32:PAPER
0:PEN 12
698 IF oy>21 AND oy<24 THEN cc=244:PAPER
 0:PEN 14
699 IF oy=24 THEN cc=243:PAPER 0:PEN 15
700 LOCATE ox,oy:PRINT CHR$(cc):RETURN
705 REM >>>>>>>> DRAW SCREEN <<<<<<<<
710 MODE 0:INK 15,24:INK 14,26:BORDER 0:
PAPER 0:CLS:PRINT CHR$(30)::INK 0,0:bord
r 0
720 PRINT:PAPER 0:PEN 15:PRINT ban1$;:PA
PER 0:PEN 10:PRINT riv$;:PAPER 0:PEN 14:
PRINT ban2$:PRINT:PRINT:PRINT:PRINT:PRIN
T:PRINT:PEN 14:PRINT ban3$;:PEN 15:PRINT
 ban4$
725 PRINT CHR$(30);"Score  High  Skipps"
730 RETURN
735 REM >>>>> MOVEMENT FROM INKEY$ <<<<
740 GOSUB 29000
745 CALL &BB03
750 IF up=1 AND ky<11 THEN k1=232:k2=233
:kdy=-2:kdx=0:sc=sc+1 ELSE IF up=1 THEN
k1=232:k2=233:kdy=-1:kdx=0:sc=sc+1
760 IF ri=1 AND kx<19 THEN k1=234:k2=235
:kdx=1:kdy=0
770 IF le=1 AND kx>1 THEN k1=236:k2=237:
kdx=-1:kdy=0
775 IF ex=1 THEN END
780 IF do=1 AND ky<11 THEN k1=238:k2=239
:kdy=2:kdx=0 ELSE IF do=1 THEN k1=238:k2
=239:kdy=1:kdx=0
```

```
781 kx=kx+kdx:ky=ky+kdy:LOCATE kx,ky:GOS
UB 30000
782 IF le=1 OR up=1 OR do=1 OR ri=1 THEN
 FOR so=300 TO 0 STEP -10:SOUND 1,so,1:N
EXT so
783 kdx=0:kdy=0:IF k=k1 THEN k=k2 ELSE k
=k1
784 GOTO 25000
790 GOSUB 693:GOSUB 690:IF ret=1 THEN GO
TO 375 ELSE GOTO 510
830 REM >>>>> CHARACTER DEFINITIONS <<<
850 SYMBOL 224,7,28,56,127,219,231,36,24
860 SYMBOL 225,128,64,32,252,218,229,38,
24
870 SYMBOL 226,0,0,0,0,255,255,60,24
880 SYMBOL 227,64,64,64,64,255,255,128,0
890 SYMBOL 228,7,9,9,63,127,127,7,3
900 SYMBOL 229,191,109,182,109,218,125,1
71,95
910 SYMBOL 230,255,118,235,182,47,254,93
,190
920 SYMBOL 231,250,213,190,91,182,109,18
2,253
930 SYMBOL 232,36,24,90,60,24,60,60,219
940 SYMBOL 233,36,24,24,60,90,189,126,36
950 SYMBOL 234,4,12,14,24,60,188,184,94
960 SYMBOL 235,8,24,28,24,60,61,58,220
970 SYMBOL 236,32,48,112,24,60,61,29,122
980 SYMBOL 237,16,24,56,24,60,188,92,59
990 SYMBOL 238,219,60,60,24,60,90,24,36
1000 SYMBOL 239,36,126,189,90,60,24,24,3
6
1010 SYMBOL 240,0,0,67,172,240,192,255,0
1020 SYMBOL 241,0,1,51,14,55,72,72,48
1030 SYMBOL 242,96,200,244,12,154,164,22
8,24
1040 SYMBOL 243,255,255,255,255,255,255,
255,255
1050 SYMBOL 244,0,127,127,127,127,127,12
7,127
1051 SYMBOL 245,170,85,170,85,170,85,170
,85
1060 addr=34999:GOSUB 29500
1065 loca=35020:GOSUB 30500
1070 RETURN
```

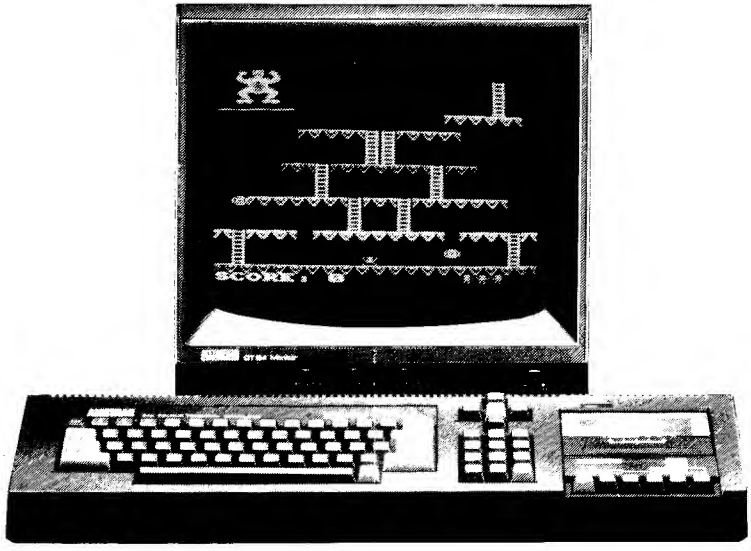
```
14990 REM >>>>> CHECK COLLISION <<<<<
25000 IF ky=4 THEN 25030
25005 IF ch=32 OR ch=244 OR ch=0 THEN 79
0
25010 IF ch=229 OR ch=230 OR ch=231 THEN
 790
25011 IF ch>231 AND ch<240 THEN 790
25015 GOTO 50500
25030 IF kx=4 THEN 50000 ELSE IF kx=8 TH
EN 50000 ELSE IF kx=12 THEN 50000 ELSE I
F kx=16 THEN 50000
25031 cc=243:ox=kx:oy=ky:GOTO 50500

39990 REM >>>>> SCORE,HIGH,SKIPPS <<<<<
40000 PRINT CHR$(30):PRINT sc;TAB(7);hi;
TAB(15);ka
40005 IF sc>hi THEN hi=sc:GOTO 40000
40010 RETURN
49990 REM >>>>> KANGAROO IN BAY <<<<<
50000 IF ch<>32 THEN 50500 ELSE k=234:GO
SUB 690:sc=sc+5:kab=kab+1:FOR n=1 TO 3:F
OR so=1 TO 200 STEP 10:SOUND 1,so,1:NEXT
 so:NEXT n
50005 IF kab=4 THEN 310 ELSE kx=10:ky=22
:GOTO 790
50490 REM >>> KANGAROO SQUASHED ! <<<
50500 GOSUB 690:SOUND 1,1000,30:SOUND 1,
3000,60:GOSUB 693:ka=ka-1:kx=10:ky=22:ox
=10:oy=22
50505 IF ka=-1 THEN GOSUB 60000:GOTO 300
50510 GOTO 370
```

Now 'MERGE' the 'INKEYS' , the

'CHAR CHECK' and the 'HIGH SCORE'

routines . . .

21
Kinkey Dong



All Action Climax

This, the third arcade style game, is loosely inspired by a group of games in which the hero must jump over obstacles as he climbs up the ladder to get higher and higher through the screens.

The evil monster that appears on each screen has in fact abducted the hero's girlfriend and locked her up. He now spends all of his time rolling large barrels down to crush the hero.

There are three different screen layouts and if you are successful then the game continues through from the first's screen to give higher scores.

Movement is by the now familiar cursor keys or your old friend the joystick. The little hero can be made to jump by pressing the copy key or the fire button.

By following the instructions contained in the chapter on subroutines it is possible to merge this program with the 'High Score' subroutine which will provide a *Kinkey Dong* Hall of Fame.

```
1 REM <<<<<< kinkey dong <> ISSI >>>>>>
2 REM
500 SYMBOL AFTER 228
510 SYMBOL 241,66,126,66,66,66,126,66,66
520 SYMBOL 242,60,66,189,165,165,189,66,
60
530 SYMBOL 243,255,165,165,36,66,66,129,
129
540 SYMBOL 244,56,60,16,44,52,56,16,28
550 SYMBOL 245,28,60,8,52,44,28,8,56
560 SYMBOL 246,186,186,146,124,56,68,130
,130
600 MEMORY 34495
750 GOSUB 22000
1000 GOSUB 20000
1100 GOSUB 19000
1500 IF lev=1 THEN GOSUB 21100
1510 IF lev=2 THEN GOSUB 21200
1520 IF lev=3 THEN GOSUB 21300
1540 GOSUB 21500
1550 x=2:y=23
1560 bx=1:by=7:bd=1:be=0
1570 cx=20:cy=7:cd=-1:ce=0
1580 IF lev=3 THEN bx=7:cx=13
1800 GOSUB 24000
1810 GOSUB 25000
1820 GOSUB 24500
1830 PEN 7
1840 LOCATE x,y:PRINT om$
1850 GOSUB 23000
1860 IF y=1 THEN GOTO 3000
1870 LOCATE x,y
1880 GOSUB 30000
1890 IF ch<>241 AND ch<>32 THEN GOTO 500
0
1900 LOCATE x,y+1
1910 GOSUB 30000
1920 IF ch=32 THEN y=y+1
1940 PEN 7
1950 LOCATE bx,by:PRINT ob$
1960 LOCATE cx,cy:PRINT oc$
2000 GOTO 1800
3000 lev=lev+1:IF lev=4 THEN lev=1
3050 sc=sc+(lev*50)
3060 LOCATE 7,25:PRINT sc
```

```
3070 GOTO 1500
5000 SOUND 2,300,75,15,0,0,30
5010 li=li-1:IF li=0 THEN GOTO 6000
5020 FOR n=1 TO li
5030 LOCATE 17+n,25
5040 NEXT n
5050 PRINT " ";
5060 GOTO 1500
6000 CLS
6005 PRINT:PEN 3
6010 PRINT "  TOUGH  MONKEYS !"
6020 PRINT " ------------------"
6030 LOCATE 3,10:PRINT "You scored ";sc;
6040 LOCATE 5,25:PRINT "Press Any Key"
6050 a$=INKEY$:IF a$="" THEN GOTO 6050
6060 SOUND 1,250,25,15
6070 GOTO 1000
9980 REM >>>>>>>>>> monster <<<<<<<<<<
10000 SYMBOL 228,0,0,16,44,62,28,56,112
10010 SYMBOL 229,129,66,126,90,126,106,8
6,60
10020 SYMBOL 230,0,0,8,52,124,56,28,14
10030 SYMBOL 231,112,121,127,63,31,7,1,1
10040 SYMBOL 232,60,255,255,255,231,215,
235,213
10050 SYMBOL 233,14,158,254,252,248,224,
128,128
10060 SYMBOL 234,1,3,15,15,31,63,62,62
10070 SYMBOL 235,171,213,235,215,235,255
,60,0
10080 SYMBOL 236,128,192,240,240,248,252
,124,124
10090 SYMBOL 237,30,15,15,3,59,127,127,5
7
10100 SYMBOL 238,0,0,0,0,129,195,195,129
10110 SYMBOL 239,120,240,240,192,220,254
,254,156
10120 PEN 15
10130 LOCATE 2,1:PRINT CHR$(228);CHR$(22
9);CHR$(230)
10140 LOCATE 2,2:PRINT CHR$(231);CHR$(23
2);CHR$(233)
10150 LOCATE 2,3:PRINT CHR$(234);CHR$(23
5);CHR$(236)
10160 LOCATE 2,4:PRINT CHR$(237);CHR$(23
```

```
8);CHR$(239)
10200 RETURN
18980 REM >>>>>>>> start game <<<<<<<<<
19000 sc=0:li=3
19010 lev=1
19020 l$=CHR$(241)
19030 b$=CHR$(242):m$=CHR$(244)
19100 RETURN
19980 REM >>>>>>>> set screen <<<<<<<<<
20000 RESTORE 20000
20010 FOR n=0 TO 15
20020 READ a
20030 INK n,a
20040 NEXT n
20050 BORDER 0:PAPER 0:PEN 1:MODE 0
20100 RETURN
20200 DATA 0,18,6,24,20,7,15,2
20210 DATA 22,12,17,9,26,3,22,13
20980 REM >>>>>>>>>> screen <<<<<<<<<<
21000 p$=STRING$(20,243)
21010 CLS:pe=1
21020 FOR n=8 TO 24 STEP 4
21025 PEN pe
21030 LOCATE 1,n:PRINT p$
21040 pe=pe+1
21070 NEXT n
21080 RETURN
21095 REM * * * one * * *
21100 GOSUB 21000
21120 FOR n=8 TO 20 STEP 4
21130 po=RND(1)*17+2
21140 IF n<24 THEN LOCATE po,n:PRINT " "
21150 lpo=RND(1)*17+2:IF lpo=po THEN GOT
O 21150
21160 IF n<24 THEN po=lpo:GOSUB 21700
21170 PEN 10
21180 NEXT n
21190 RETURN
21195 REM * * * two * * *
21200 GOSUB 21000
21205 PEN 12
21210 FOR n=13 TO 24
21220 LOCATE 7,n:PRINT STRING$(8,207)
21230 NEXT n
21240 LOCATE 7,19:PRINT STRING$(8,32)
```

```
21250 LOCATE 7,23:PRINT STRING$(8,32)
21260 LOCATE 7,15:PRINT STRING$(8,32)
21270 po=8:n=8:GOSUB 21700:po=18:n=12:GO
SUB 21700
21275 po=3:n=16:GOSUB 21700:po=18:n=20:G
OSUB 21700
21280 PEN 10
21292 RETURN
21295 REM * * * three * * *
21300 GOSUB 21000
21310 LOCATE 1,8:PRINT STRING$(5,32)
21320 LOCATE 16,8:PRINT STRING$(5,32)
21330 LOCATE 1,12:PRINT STRING$(4,32)
21340 LOCATE 17,12:PRINT STRING$(4,32)
21350 LOCATE 1,16:PRINT STRING$(2,32)
21360 LOCATE 19,16:PRINT STRING$(2,32)
21370 po=7:n=12:GOSUB 21700
21380 po=14:n=12:GOSUB 21700
21390 po=2:n=20:GOSUB 21700
21400 po=19:n=20:GOSUB 21700
21410 po=9:n=16:GOSUB 21700
21420 po=12:n=16:GOSUB 21700
21430 po=10:n=8:GOSUB 21700
21440 po=11:n=8:GOSUB 21700
21450 LOCATE 6,20:PRINT " ":LOCATE 15,20
:PRINT " "
21455 PEN 10
21480 RETURN
21490 REM * * * top * * *
21500 LOCATE 1,25
21505 PEN 10
21510 PRINT "SCORE:";sc
21520 LOCATE 16,25
21525 PEN 6
21530 FOR n=1 TO li
21540 PRINT CHR$(248);
21550 NEXT n
21555 PEN 7
21560 LOCATE 15,6:PRINT STRING$(5,243)
21570 PLOT 0,325:DRAW 150,325
21580 po=18:n=2:GOSUB 21700
21650 GOSUB 10000
21680 RETURN
21690 REM * * * ladder * * *
21700 FOR z=0 TO 3
```

```
21705 PEN 7
21710 LOCATE po,n+z
21720 PRINT l$
21730 NEXT z
21740 RETURN
21980 REM >>>>>>>> initialize <<<<<<<<<
22000 addr=34999:GOSUB 29500
22010 loca=35024:GOSUB 30500
22150 RETURN
22980 REM >>>>>>>>>> move <<<<<<<<<<<<
23000 GOSUB 29000
23050 IF le=1 AND x>2 THEN x=x-1:m$=CHR$
(245)
23060 IF ri=1 AND x<19 THEN x=x+1:m$=CHR
$(244)
23070 IF up=1 THEN m$=CHR$(246):GOTO 233
00
23080 IF do=1 THEN m$=CHR$(246):GOTO 234
00
23090 IF fi=1 THEN GOTO 23500
23250 RETURN
23300 LOCATE x,y-1
23305 GOSUB 30000
23310 IF ch<>241 THEN GOTO 23350
23320 y=y-1:GOTO 23080
23350 LOCATE x,y+1
23360 GOSUB 30000
23370 IF ch=241 THEN GOTO 23320
23380 GOTO 23080
23400 LOCATE x,y+1
23405 GOSUB 30000
23410 IF ch<>241 THEN GOTO 23090
23420 y=y-1:GOTO 23090
23500 IF m$=CHR$(245) THEN xd=-1 ELSE xd
=1
23510 IF x<2 OR x>19 THEN xd=0
23520 IF m$=CHR$(246) THEN xd=0
23530 LOCATE bx,by:PRINT ob$
23540 LOCATE cx,cy:PRINT oc$
23610 RESTORE 23750
23650 FOR n=1 TO 6
23660 READ a
23670 y=y+a:GOSUB 23850
23680 IF plat=1 THEN n=11
23690 NEXT n
```

```
23700 GOSUB 24500
23710 RETURN
23750 DATA -1,-1,0,0,1,1
23850 x=x+xd:plat=0
23860 LOCATE x,y+1
23870 GOSUB 30000
23880 IF ch<>32 THEN plat=1:RETURN
23890 IF x<2 THEN x=2
23895 IF x>19 THEN x=19
23900 GOSUB 24000
23910 GOSUB 24500
23920 SOUND 2,100,2,15
23925 PEN 7
23930 LOCATE bx,by:PRINT ob$
23935 LOCATE cx,cy:PRINT oc$
23940 GOSUB 25000
23950 LOCATE x,y:PRINT om$
23960 RETURN
23980 REM >>>>>>>>>> print <<<<<<<<<<<
23995 REM * * * man * * *
24000 LOCATE x,y:GOSUB 30000
24010 om$=CHR$(ch)
24020 PEN 2
24050 PRINT m$
24100 RETURN
24490 REM * * * barrels * * *
24500 LOCATE bx,by:GOSUB 30000
24510 ob$=CHR$(ch)
24520 LOCATE cx,cy:GOSUB 30000
24530 oc$=CHR$(ch)
24540 PEN 3
24550 LOCATE bx,by
24560 PRINT b$
24570 LOCATE cx,cy
24580 PRINT b$
24600 RETURN
24980 REM >>>>>>>>>> barrel <<<<<<<<<<
25000 a=RND(1)*10
25005 LOCATE bx,by+1
25010 GOSUB 30000
25020 IF ch=32 OR ch=241 AND a>5 THEN be
=1
25030 IF ch=243 THEN be=0
25040 by=by+be:IF be=0 THEN bx=bx+bd
25050 LOCATE cx,cy+1
```

```
25060 GOSUB 30000
25070 IF ch=32 OR ch=241 AND a>5 THEN ce
=1
25080 IF ch=243 THEN ce=0
25090 cy=cy+ce:IF ce=0 THEN cx=cx+cd
25100 IF bx<2 THEN bd=1
25105 IF bx>19 THEN bd=-1
25110 IF cx<2 THEN cd=1
25115 IF cx>19 THEN cd=-1
25130 IF (bx<2 OR bx>19) AND by=23 THEN
bx=1:bd=1:by=7:IF lev=3 THEN bx=7
25140 IF (cx<2 OR cx>19) AND cy=23 THEN
cx=20:cd=-1:cy=7:IF lev=3 THEN cx=13
25250 RETURN
```

Now 'MERGE' the 'INKEYS' and the

'CHAR CHECK' routines. . .

22
Word Splash

Spell or Swim

This game is a more humane, more fun version of the well known hangman game. If the player cannot guess the word in time, then all the wrong letters push a character into the water.

Words can be up to eight letters long but should not contain hyphens. The listing has about 60 words already included. More words may be added or the existing ones changed.

If a letter is guessed, it appears at each correct position in the word. If it is not right then the poor victim is pushed one space along. Should the word be completed in time then the little figure pushes all the wrong letters back and a new mystery word is offered.

The word is not shown if it is not guessed and since words are randomly selected it may be possible to have the same word again later.

The program is very entertaining, educational and can be surprisingly addictive!

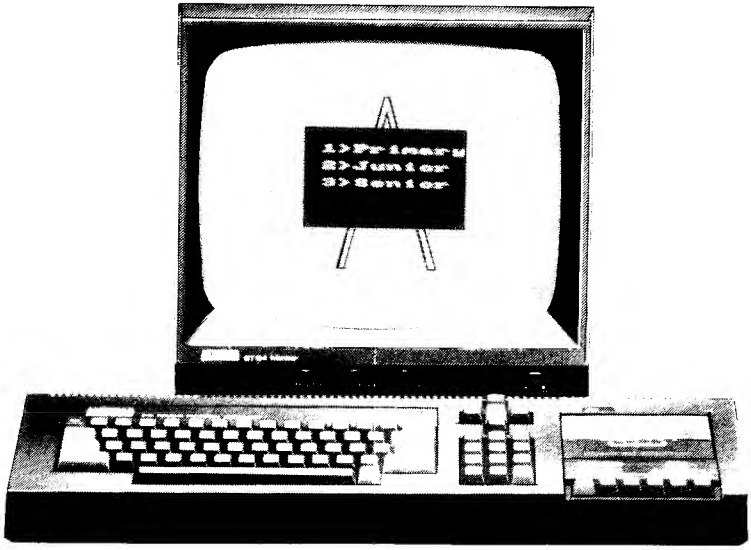
```
1 REM <<<<<< word splash <> ISSI >>>>>>
2 REM
500 RANDOMIZE TIME
1000 GOSUB 21000
1500 RANDOMIZE TIME
1510 w=INT(RND(1)*60)+1
1520 w$=d$(w)
1530 GOSUB 20000
1600 GOSUB 24000
1610 flag=0
1650 IF t$=w$ THEN GOTO 3000
1660 IF LEN(g$)=10 THEN GOTO 2000
1670 GOTO 1600
1990 REM * * * lose * * *
2000 GOSUB 23000
2010 RESTORE 2000
2020 FOR n=1 TO 10
2030 READ a,b
2040 SOUND 2,a,b,15
2050 NEXT n
2060 DATA 253,20,284,20,319,50,284,20
2070 DATA 319,20,358,50,319,20,358,20
2080 DATA 379,50,358,50
2400 CLS
2410 LOCATE 4,3:PRINT "TOUGH LUCK !"
2500 GOTO 3500
2990 REM * * * win * * *
3000 RESTORE 3000
3010 FOR n=1 TO 16
3020 READ a,b
3030 SOUND 2,a,b,15
3040 NEXT n
3050 DATA 319,20,284,20,253,20,239,80
3060 DATA 319,20,284,20,253,20,239,80
3070 DATA 319,20,284,20,253,20,239,40
3080 DATA 284,40,358,40,284,40,319,80
3220 IF g$="" THEN GOTO 3480
3230 FOR n=LEN(g$) TO 1 STEP-1
3240 g$=RIGHT$(g$,n-1)
3260 GOSUB 24110:PRINT " "
3270 SOUND 1,n*20,2,15
3280 FOR p=1 TO 500
3290 NEXT p
3300 NEXT n
3480 CLS
```

```
3490 LOCATE 2,3:PRINT "CONGRATULATIONS !
"
3500 LOCATE 3,20:PRINT "Press Any Key"
3520 a$=INKEY$
3530 IF a$="" THEN GOTO 3520
3540 GOTO 1500
19980 REM >>>>>>>>>> screen <<<<<<<<<<
20000 RESTORE 20000
20010 FOR n=0 TO 15
20020 READ a
20030 INK n,a
20040 NEXT n
20050 DATA 0,20,2,18,6,24,26,0
20060 DATA 0,0,0,0,0,0,0,0
20100 BORDER 0:PAPER 0:PEN 1:MODE 0
20110 LOCATE 5,1:PRINT "WORD  SPLASH"
20120 a$=CHR$(196)+CHR$(198):a$=a$+a$+a$
+a$+a$+a$+a$+a$
20130 PEN 6
20140 LOCATE 4,2:PRINT a$
20150 PEN 4
20160 LOCATE 1,11:PRINT STRING$(10,207)+
CHR$(223)
20170 LOCATE 10,12:PRINT CHR$(221)
20180 LOCATE 11,12:PRINT CHR$(207)
20190 LOCATE 12,12:PRINT CHR$(223)
20210 FOR n=1 TO 10
20220 LOCATE 12,14+n
20230 PAPER 2:PEN 3
20235 IF n<3 THEN PAPER 0
20240 PRINT CHR$(211);STRING$(6,32);CHR$
(209);
20250 NEXT n
20260 LOCATE 12,25:PRINT CHR$(205);STRIN
G$(6,210);CHR$(204);
20300 RESTORE 20300
20310 FOR n=8 TO 14
20320 PAPER 0:PEN n
20330 READ a,b
20340 LOCATE a,b
20350 PRINT CHR$(249);
20360 NEXT n
20370 DATA 11,10,12,11,13,12,14,13
20380 DATA 15,14,16,15,16,16
20400 PLOT 510,150,7:DRAW 560,160
```

```
20410 PLOT 478,150:DRAW 420,160
20420 PLOT 510,155:DRAW 540,180
20430 PLOT 478,155:DRAW 448,180
20440 PLOT 520,147:DRAW 590,150
20450 PLOT 468,147:DRAW 398,150
20490 PEN 5
20500 LOCATE 1,13:PRINT "WORD:-"
20510 LOCATE 2,17:t$="":g$=""
20520 FOR n=1 TO LEN(w$)
20530 PRINT "-";:t$=t$+" "
20540 NEXT n
20550 PEN 3
20560 LOCATE 1,10:PRINT CHR$(248)
20580 LOCATE 2,20:PRINT "GUESS ?"
20750 RETURN
20980 REM >>>>>>>>> initialize <<<<<<<<<
21000 RESTORE 30000
21010 DIM d$(60)
21020 FOR n=1 TO 60
21030 READ d$(n)
21040 NEXT n
21250 RETURN
22980 REM >>>>>>>>>>> splash! <<<<<<<<<<
23000 LOCATE 11,10:PRINT " "
23010 FOR n=8 TO 14
23020 INK n,24
23030 SOUND 2,300-n*10,25,15
23040 FOR q=1 TO 250
23050 NEXT q
23060 INK n,0
23070 NEXT n:INK 14,24
23080 INK 7,26:SOUND 2,300,50,15,0,0,31
23090 FOR n=1 TO 250
23100 NEXT n
23110 INK 7,0:INK 14,0
23250 RETURN
23980 REM >>>>>>>>>>> input <<<<<<<<<<<
24000 a$=INKEY$:a$=UPPER$(a$)
24010 IF a$<"A" OR a$>"Z" THEN GOTO 2400
0
24020 SOUND 2,200,10,15
24030 flag=0:PEN 4
24040 FOR n=1 TO LEN(w$)
24050 IF MID$(w$,n,1)=a$ THEN LOCATE 1+n
,16:PRINT a$:MID$(t$,n,1)=a$:flag=1
```

```
24060 NEXT n
24070 IF flag=0 THEN GOTO 24100
24080 SOUND 2,150,50,15
24090 RETURN
24100 g$=a$+g$
24110 LOCATE 1,10:PEN 4
24120 IF LEN(g$)/2=INT(LEN(g$)/2) THEN a
=250 ELSE a=251
24130 PEN 3:PRINT g$;CHR$(a);
24140 RETURN
29980 REM >>>>>>>>>>> data <<<<<<<<<<<
30000 DATA ACUTE,AGILE,BROKE,CANDY,COUGH
,DRYLY,GNOME,MISTY,NYMPH,PYGMY
30010 DATA BARLEY,BODKIN,CEPTIC,DEVOUR,E
IGHTY,EXOTIC,FLINCH,GLORIA,HYMN,LYRICS
30020 DATA METHYL,SPYING,SYRUP,YOGHURT,V
ISUAL,WOBBLE,ANCHOVY,RELAX,RHYTHM,RHUBAR
B
30030 DATA SKETCH,FASCIST,BORING,FATIGUE
,FROWN,TRUANT,LEMMING,ZINC,GLAMOUR,HYDRA
TED
30040 DATA KETCHUP,MYSTIC,NOISY,OCTOPUS,
SQUID,RAMBLER,SATCHEL,SEXTANT,YULETIDE,S
WINDLE
30050 DATA PLAYER,DOCKER,DESTINY,FUTURE,
LENTIL,FOREIGN,LOGICAL,WINTER,THIRST,NAI
VE
```

23
Sum Fun



Ten out of Ten

This single program can offer questions for 'primary', 'junior' and 'senior' levels (that is everyone). The question can be of any one type, selected from plus, minus, multiply and divide. The questions and answers are drawn to a blackboard to help make the presentation attractive.

Selecting the primary level will set up problems which involve counting two sets of 'little men' to produce a total. The junior level provides options for the four types of arithmetic and produces questions which have whole number answers. It should be possible for most adults and older children to work these out mentally.

The senior level would require a very strong mental arithmetic ability and is designed to set problems which can be worked out using pencil and paper or even a calculator.

The answers are required to be typed in and then sent to the computer using the 'Enter' key. This means that mistyped answers can be corrected if spotted before the 'Enter' key is pressed. The

program will allow up to two wrong trys before revealing the correct answer.

At the end of ten questions the computer will print up a 'report' showing the number of correct answers and a percentage score.

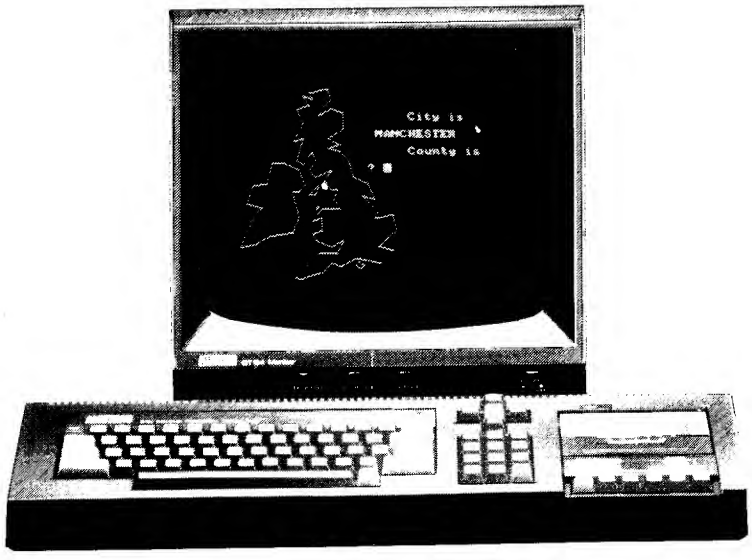
After "RENUM 40000,5" the game *Rainbow Breakout* may be merged with this program to provide a fun reward for getting ten out of ten.

```
1 REM <<<<<<<< sum fun <> ISSI >>>>>>>>
2 REM
1200 GOSUB 20000
1300 LOCATE #1,2,3:PRINT #1,"1>Primary"
1310 LOCATE #1,2,5:PRINT #1,"2>Junior"
1320 LOCATE #1,2,7:PRINT #1,"3>Senior"
1330 a$=INKEY$
1340 IF a$<"1" OR a$>"3" THEN GOTO 1330
1350 SOUND 2,300,50,15
1360 lev=VAL(a$)
1390 GOTO 30000
19980 REM >>>>>>>>>> screen <<<<<<<<<<
20000 RESTORE 20000
20010 FOR n=0 TO 7
20020 READ a
20030 INK n,a
20040 NEXT n
20050 DATA 2,0,26,24,6,18,7,13
20060 BORDER 2:PAPER 0:PEN 6:MODE 0
20070 FOR n=320 TO 335 STEP 4
20080 PLOT n,380,6:DRAWR 80,-330
20090 NEXT n
20100 FOR n=320 TO 305 STEP -4
20110 PLOT n,380,6:DRAWR -80,-330
20120 NEXT n
20130 WINDOW #1,6,15,6,17
20140 WINDOW #2,8,13,14,14
20150 PAPER #1,1:PEN #1,2:CLS #1
20160 PAPER #2,1:PEN #2,2:CLS #2
20250 RETURN
20980 REM >>>>>>>>>> input <<<<<<<<<<
21000 INPUT #2,g$
21010 IF g$="" THEN GOTO 21000
21020 cor=1
21030 IF VAL(g$)<>ans THEN cor=0
21040 RETURN
21490 REM * * * guess * * *
```

```
21500 FOR g=1 TO 2
21510 GOSUB 21000
21520 IF cor=1 THEN GOTO 21560
21530 SOUND 2,500,40,15:SOUND 2,750,50,1
5
21540 NEXT g
21542 RESTORE 21542
21544 FOR n=1 TO 9
21546 READ a:SOUND 2,a,20,15
21548 NEXT n
21550 DATA 200,250,300,250,300,350,300,3
50,400
21552 SOUND 2,600,75,15
21555 GOTO 21590
21560 SOUND 2,600,30,15:SOUND 2,400,30,1
5:SOUND 2,200,30,15
21570 g=2
21580 tot=tot+1
21590 CLS #1:CLS #2
21600 RETURN
21980 REM >>>>>>>>>>> type <<<<<<<<<<<<
22000 CLS #1:CLS #2
22010 LOCATE #1,2,2:PRINT #1,"1> '+'"
22020 LOCATE #1,2,4:PRINT #1,"2> '-'"
22030 LOCATE #1,2,6:PRINT #1,"3> 'x'"
22040 LOCATE #1,2,8:PRINT #1,"4> '";CHR$
(172);"'"
22050 a$=INKEY$
22060 IF a$<"1" OR a$>"4" THEN GOTO 2205
0
22070 z=VAL(a$)
22080 DEF FN sum=a+b:s$="+"
22090 IF z=2 THEN DEF FN sum=a-b:s$="-"
22100 IF z=3 THEN DEF FN sum=a*b:s$="x"
22110 IF z=4 THEN DEF FN sum=a/b:s$=CHR$
(172):div=1
22200 RETURN
22980 REM >>>>>>>>>>> init <<<<<<<<<<<<
23000 lim=8:div=0
23010 IF lev=2 THEN lim=12:GOSUB 22000
23020 IF lev=3 THEN lim=25:GOSUB 22000
23030 CLS #1:CLS #2
23040 tot=0
23100 RETURN
23980 REM >>>>>>>>> numbers <<<<<<<<<<
```

```
24000 RANDOMIZE TIME
24010 a=INT(RND(1)*lim)+1
24020 b=INT(RND(1)*lim)+1
24030 IF div=1 THEN a=a*b
24035 IF lev=2 THEN div=0
24040 CLS #1:CLS #2
24050 IF lev=1 THEN GOTO 24150
24060 LOCATE #1,1,3:PRINT #1,a;s$;b
24070 ans=FN sum
24100 RETURN
24150 LOCATE #1,2,3:PRINT #1,STRING$(a,2
50):LOCATE #1,5,5:PRINT #1,"+"
24160 LOCATE #1,2,7:PRINT #1,STRING$(b,2
50)
24190 ans=a+b
24200 RETURN
29980 REM >>>>>>>>> main loop <<<<<<<<<
30000 GOSUB 23000
30010 FOR p=1 TO 10
30040 GOSUB 24000
30050 GOSUB 21500
30100 NEXT p
30150 CLS #1:CLS #2
30160 PRINT #1,"  REPORT."
30170 PRINT #1," ---------"
30180 LOCATE #1,2,5:PRINT #1,tot;"were"
30190 LOCATE #1,2,7:PRINT #1,"correct."
30200 LOCATE #1,3,9:PRINT #1,tot/10*100
"%"
30250 LOCATE 4,25:PRINT "Press any Key"
30260 a$=INKEY$
30270 IF a$="" THEN GOTO 30260
30280 LOCATE 4,25:PRINT SPC(13)
30290 CLS #1:CLS #2
30300 GOSUB 40000
30310 GOTO 1300
40000 RETURN
```

24
Today England



Tomorrow the World

This is a self-contained geographic quiz based upon the map of Great Britain. The game is very straightforward and consists of questions which relate to key towns or cities, and the county to which they belong. Options are given to guess the county, the town or both. A point is plotted on the map to indicate the location of the town or city.

The central core of the program could be developed to produce either a more detailed map or as the heading suggests, the world could be plotted next!

The *Picture Builder* program later in this book will be found helpful to identify the points to be plotted. The map used in this program was produced by taping an outline of Great Britain onto the screen. This was on a clear acetate sheet, so it was possible to move the cursor to each point, note the x,y co-ordinates and then move to the next point. The result can be as detailed as the mode selected and your patience will allow.

Another way to develop this program would be to extend the data to include more or all counties and the county towns. In fact you

could find that the research of the facts to put into the listing is even more educational then the program itself!

```
10 REM <<<< TODAY ENGLAND , >>>>>>>>
11 REM << TOMMORROW THE WORLD >>>>>>
15 REM <<<<< ANDY / JIM       >>>>>>>
16 REM
20 GOSUB 1500
30 MODE 1:LOCATE 15,1:PRINT "Today Engla
nd"
35 INK 0,0:BORDER 0
40 sc=0
50 LOCATE 12,3:PRINT "Tommorrow the Worl
d"
60 LOCATE 2,6:PRINT "Choose option:"
70 LOCATE 10,8:PRINT "1....Guess countie
s"
80 LOCATE 10,9:PRINT "2....Guess cities"
90 LOCATE 10,10:PRINT "3....Guess both"
100 a$=INKEY$
110 IF a$="1" THEN op=1:GOTO 150
120 IF a$="2" THEN op=2:GOTO 150
130 IF a$="3" THEN op=3:GOTO 150
140 GOTO 100
150 GOSUB 900
160 RESTORE 1270
165 IF op=3 THEN 630
210 REM >>> GUESS CITY / COUNTY <<<<<
220 FOR c=1 TO 16
230 READ n$,x,y,co$
240 PLOT x,y,3
250 IF op=1 THEN LOCATE 25,5:PRINT "City
 is "
260 IF op=1 THEN LOCATE 21,7:PRINT n$
270 IF op=2 THEN LOCATE 25,5:PRINT "Coun
ty is "
280 IF op=2 THEN LOCATE 21,7:PRINT co$
290 tr=3
300 IF op=1 THEN LOCATE 25,9:PRINT "Coun
ty is" ELSE IF op=2 THEN LOCATE 25,9:PRI
NT "City is"
310 IF op=1 THEN INPUT #3,c$:c$=UPPER$(c
$)
320 IF op=2 THEN INPUT #3,ci$:ci$=UPPER$
(ci$)
```

```
330 IF op=1 AND c$=co$ THEN 420
340 IF op=2 AND ci$=n$ THEN 420
350 tr=tr-1:LOCATE 26,20:PRINT "Wrong";t
r;" tries"
360 LOCATE 30,22:PRINT "left."
370 GOSUB 1500
380 LOCATE 26,20:PRINT SPC(14)
390 LOCATE 30,22:PRINT SPC(6)
400 IF tr=0 THEN 5000
410 GOTO 300
420 GOSUB 1440
430 LOCATE 26,10:PRINT "Correct."
440 GOSUB 1500
450 sc=sc+tr
460 PLOT x,y,0
470 PRINT #3
480 PLOT 0,0,1
490 GOSUB 1440
500 NEXT c
510 CLS:PRINT:PRINT "      You scored";sc
;" points."
520 PRINT:PRINT:PRINT"     Press 'SPACE'
to start again"
530 PRINT:PRINT"           or 'Q' to quit
."
540 a$=INKEY$:a$=UPPER$(a$)
550 IF a$=" "THEN 30
560 IF a$="Q" THEN END
570 GOTO 540
610 REM >>>>>> GUESS CITY & COUNTY <<<
620 REM
630 FOR c=1 TO 16
640 READ n$,x,y,c$:PLOT x,y,3
650 LOCATE 25,9:PRINT "City is"
660 tr=3
670 INPUT #3,ci$
680 IF ci$=n$ THEN GOSUB 1440:LOCATE 25,
10:PRINT "Correct":GOSUB 1500:sc=sc+tr:G
OSUB 1440:GOTO 750
690 tr=tr-1
700 LOCATE 26,20:PRINT "Wrong";tr;" trie
s"
710 LOCATE 30,22:PRINT "left."
720 GOSUB 1500
725 LOCATE 26,20:PRINT SPC(14)
```

```
726 LOCATE 30,22:PRINT SPC(6)
727 IF tr=0 THEN 730 ELSE GOTO 670
730 LOCATE 25,1:PRINT "CITY IS"
740 LOCATE 20,3:PRINT n$
750 LOCATE 25,9:PRINT "County is"
760 tr=3
770 INPUT #3,co$
780 IF co$=c$ THEN GOSUB 1440:LOCATE 25,
10:PRINT "Correct":GOSUB 1500:sc=sc+tr:G
OSUB 1440:GOTO 841
790 tr=tr-1
800 LOCATE 26,20:PRINT "Wrong";tr;" trie
s"
810 LOCATE 30,22:PRINT "left."
820 GOSUB 1500
821 LOCATE 26,20:PRINT SPC(14)
822 LOCATE 30,22:PRINT SPC(6)
825 IF tr=0 THEN 830 ELSE GOTO 770
830 LOCATE 25,9:PRINT "                "
832 PRINT #3
835 LOCATE 25,6:PRINT "COUNTY IS"
840 LOCATE 20,8:PRINT c$
841 GOSUB 1500
842 GOSUB 1440
843 PLOT x,y,0
845 NEXT c:GOTO 510
890 REM >>>>>>>>> DRAW MAP <<<<<<<<<
900 MODE 1:WINDOW #3,20,40,11,11
910 PLOT 0,0,1
920 INK 3,0,26
930 RESTORE 1130
940 FOR a=1 TO 59
950 READ x,y
960 IF a=1 THEN PLOT x,y ELSE DRAW x,y
970 NEXT a
980 FOR a=1 TO 17
990 READ x,y
1000 IF a=1 THEN PLOT x,y ELSE DRAW x,y
1010 NEXT a
1020 FOR a=1 TO 4:READ x,y
1030 IF a=1 THEN PLOT x,y ELSE DRAW x,y
1040 NEXT a
1050 FOR a=1 TO 4:READ x,y
1060 IF a=1 THEN PLOT x,Y ELSE DRAW x,y
1070 NEXT a
```

```
1080 FOR a=1 TO 5:READ x,y
1090 IF a=1 THEN PLOT x,y ELSE DRAW x,y
1100 NEXT a
1110 RETURN
1115 REM >>>> MAPS DATA <<<<
1120 REM Mainland Britain
1130 DATA 192,371,228,377,234,359,210,32
9,234,341,258,335,258,305,228,287,252,27
5
1140 DATA 228,269,252,263,270,245,276,21
5,306,203,318,173,294,179,324,149,318,13
7,354,143,360,113
1150 DATA 324,83,354,77,330,53,282,61,25
3,47,234,53,216,35,174,23,165,24,210,71,
246,71,258,95,234,83
1160 DATA 204,95,198,107,216,113,216,137
,198,131,216,155,252,155,240,173,246,191
,234,191,222,209
1170 DATA 234,227,204,215,192,227,204,24
5,198,263,180,239,168,245,180,287,162,28
1,180,317,168,341,192,347,180,365,198,35
9,192,371
1180 REM Ireland
1190 DATA 142,241,172,229,184,199,172,18
7,160,187,172,139,160,109,118,103,82,85,
58,80,80,130,100,157,70,163,82,199,110,2
00,120,240,142,241
1200 REM Isle of Man
1210 DATA 192,181,204,181,210,199,192,18
1
1220 REM Anglesey
1230 DATA 206,157,206,151,210,160,206,15
7
1240 REM Isle of Wight
1250 DATA 286,55,280,49,292,43,298,49,28
6,55
1260 REM cities
1270 DATA MANCHESTER,258,171,"GREATER MA
NCHESTER"
1280 DATA CHESTER,252,149,"CHESHIRE"
1290 DATA LEEDS,282,175,"YORKSHIRE"
1300 DATA HULL,308,181,"HUMBERSIDE"
1310 DATA LIVERPOOL,250,163,"MERSEYSIDE"
1320 DATA BIRMINGHAM,274,119,"WEST MIDLA
NDS"
```

```
1330 DATA LONDON,318,85,"GREATER LONDON"
1340 DATA BRISTOL,260,83,"AVON"
1350 DATA CARDIFF,244,95,"MID GLAMORGAN"
1360 DATA CAMBRIDGE,322,115,"CAMBRIDGESH
IRE"
1370 DATA GLASGOW,216,261,"STRATHCLYDE"
1380 DATA EDINBURGH,238,261,"LOTHIAN"
1390 DATA ABERDEEN,250,315,"GRAMPIAN"
1400 DATA NEWCASTLE,266,223,"NORTHUMBERL
AND"
1410 DATA BELFAST,170,200,"ARMAGH"
1420 DATA DUBLIN,160,160,"DUBLIN"
1430 REM > CLEAR RIGHT SIDE OF SCREEN <
1440 FOR q=1 TO 24
1450 LOCATE 25,q:PRINT SPC(15)
1460 NEXT q
1465 s$=STRING$(6,32)
1470 LOCATE 20,11:PRINT s$
1480 LOCATE 20,7:PRINT s$
1485 LOCATE 20,3:PRINT s$
1486 LOCATE 20,8:PRINT s$
1490 RETURN
1500 FOR t=1 TO 3000:NEXT t:RETURN
4990 REM <<<<< NO TRIES LEFT >>>>>
5000 GOSUB 1440
5005 LOCATE 25,3
5010 PRINT "CITY IS "
5015 LOCATE 21,5
5020 PRINT n$
5025 LOCATE 25,10
5030 PRINT "COUNTY IS "
5035 LOCATE 21,12
5040 PRINT co$
5045 GOSUB 1500
5050 GOSUB 1440
5055 LOCATE 21,5:PRINT SPC(6)
5060 LOCATE 21,12:PRINT SPC(6)
5065 GOTO 460
```

25
Character Builder

The Create Your Own Monster Kit

Character building can be as much fun as playing a game. This 'utility' enables 32 of the Amstrad's characters to be defined easily and then butted together to form more complex pictures.

The built-in characters from 224 to 255 are first displayed and then the opportunity is given to adapt them, swap them, or create completely new ones. The definition values can be noted down and then used in your own programs using the 'Symbol' command. Complete sets can be saved on tape to produce a library of useful characters.

Although many of the programs in this book have had characters designed for them, others use standard characters. With this character definer you can dramatically improve the appearance of these games. You may even wish to change all the ones in the book!

Because of the wide range of features in this editor, it may take a little time learning how to drive it. The following list of commands will provide a handy reference guide during your 'Learner' period.

Character builder reference guide

After loading, the grid starts with the first character picked up. This may then be Edited, Cleared, or any of the following can be selected by pressing the first letter of the option.

Clear. This clears the grid completely.

Pick up. Asks for character number, then as the last digit is pressed that character will be transferred to the grid. It can then be Edited, Changed, etc.

Edit. Places the flashing cursor at the top left of the grid. It can then be moved around using cursor or joystick. Fire or copy key will set or unset a point. When finished 'Enter' is pressed. Then the number is required to transfer the pattern to.

Numbers. Asks for character to which the pattern will be assigned and then requires eight values, corresponding to the total for each line, starting at the top. This then generates the image directly. 'Enter' terminates this option.

Mirror. Asks for character number then asks whether horizontal (swap left to right) or vertical (turn upside down). The result is shown in the grid. To store the new character select 'Edit' followed by 'Enter' and then the character number for allocation.

Vert shift. Asks 'up' or 'down'. Moves the definition up or down one row with wraparound.

Hor shift. As for Vert shift, but left or right. Wrapped lines may be edited out.

X-change. Asks for two characters and then swaps their definitions.

Build. Removes the grid and provides a 14×10 character area in which to test how the patterns look when fitted together. The character number is requested and then that pattern may be placed anywhere using the cursor pad or the joystick. The fire button or copy key fixes a character. A new character is selected by firmly pressing 'space' and then entering the character number required. 'Enter' leaves Build and returns to grid.

Save. Asks for a file name to save with, next 'Play' and 'Record' should be pressed, then 'Enter' will save the current set.

Load. Asks for a file name to load, or if just 'Enter' is pressed the first file found will be loaded.

Note that in this program 'Escape' is disabled to prevent loss of data. Switch off to end.

```
1 REM <<< char builder <^> ISSI/JIM >>>
2 REM
200 CALL &BB03
500 SPEED WRITE 1
1000 GOSUB 21000
1010 GOSUB 20000
1020 char=1:GOSUB 22000
1050 MEMORY 34995
5010 REM >>>>>>>>> main loop <<<<<<<<<
5030 a$=INKEY$:a$=UPPER$(a$)
5040 IF a$="" THEN GOTO 5030
5050 IF a$="C" THEN GOSUB 10000
5060 IF a$="P" THEN GOSUB 11000
5070 IF a$="E" THEN GOSUB 12000
5080 IF a$="N" THEN GOSUB 13000
5090 IF a$="M" THEN GOSUB 14000
5100 IF a$="V" THEN GOSUB 15000
5110 IF a$="H" THEN GOSUB 16000
5120 IF a$="X" THEN GOSUB 17000
5130 IF a$="S" THEN GOSUB 18000
5140 IF a$="L" THEN GOSUB 19000
5150 IF a$="B" THEN GOSUB 19500
5170 GOTO 5030
9980 REM >>>>>>>>> clear grid <<<<<<<<<
10000 PRINT CHR$(23);CHR$(0);
10040 y=398
10045 b$=SPACE$(15)
10050 FOR n=1 TO 10
10060 PLOT 382,y,0
10070 PRINT #1,b$
10080 y=y-16
10090 NEXT n
10100 GOSUB 25000
10110 PRINT CHR$(23);CHR$(1);
10120 RETURN
10980 REM >>>>>>>>>> pick-up <<<<<<<<<<
11000 GOSUB 10000
11010 GOSUB 24000
11020 GOSUB 22000
11060 RETURN
11980 REM >>>>>>>>> edit grid <<<<<<<<<
12000 x=1:y=1
12050 PRINT CHR$(23);CHR$(1);
12060 PLOT 382,382,3
12100 x1=366+x*16:y1=398-y*16
```

```
12110 MOVE x1,y1:PRINT #1,CHR$(159);
12120 GOSUB 29000
12130 IF le=1 AND x>1 THEN x=x-1
12140 IF ri=1 AND x<8 THEN x=x+1
12150 IF up=1 AND y>1 THEN y=y-1
12160 IF do=1 AND y<8 THEN y=y+1
12170 MOVE x1,y1:PRINT #1,CHR$(159);
12180 IF q=1 THEN GOTO 12300
12190 IF fi=1 THEN MOVE x1,y1:PRINT #1,C
HR$(143);
12260 GOTO 12100
12290 REM * * * quit * * *
12300 GOSUB 24000
12310 b=374
12320 FOR n=1 TO 8
12330 a=390
12340 x$="        "
12350 FOR m=1 TO 8
12360 IF TEST (a,b)<>0 THEN MID$(x$,m,1)
=CHR$(143)
12370 a=a+16
12380 NEXT m
12390 GOSUB 28000
12400 d(char,n)=x
12410 b=b-16
12420 NEXT n
12430 GOSUB 14560
12440 RETURN
12980 REM >>>>>>>> value entry <<<<<<<<
13000 GOSUB 24000
13040 FOR n=1 TO 8
13050 CLS #2
13060 PRINT #2,"value ";n
13070 INPUT #2,va
13090 IF va<0 OR va>255 THEN GOTO 13080
13100 d(char,n)=va
13110 NEXT n
13160 CLS #2
13170 GOSUB 14560
13180 RETURN
13980 REM >>>>>>>>>> mirror <<<<<<<<<<
14000 GOSUB 24000
14050 PRINT #2,"(V)ert/(H)ori ?"
14060 b$=INKEY$:b$=UPPER$(b$)
14070 IF b$="V" THEN CLS #2:GOTO 14500
```

```
14080 IF b$="H" THEN CLS #2:GOTO 14100
14090 GOTO 14060
14099 REM * * * HORIZONTAL * * *
14100 FOR n=1 TO 8
14110 x=d(char,n):GOSUB 23000
14120 y$="00000000"
14130 FOR m=1 TO 8
14140 MID$(y$,m,1)=MID$(x$,9-m,1)
14150 NEXT m
14160 x$=y$:GOSUB 28000:d(char,n)=x
14170 NEXT n
14180 GOTO 14560
14490 REM * * * VERTICAL * * *
14500 FOR n=1 TO 8:f(n)=d(char,n):NEXT n

14510 FOR n=8 TO 1 STEP -1
14520 d(char,n)=f(9-n)
14530 NEXT n
14560 x=128:GOSUB 26000
14570 GOSUB 27000
14580 GOSUB 10000
14590 GOSUB 22000
14600 x=128:GOSUB 26000
14610 RETURN
14980 REM >>>>>>> vert. shift <<<<<<<
15000 GOSUB 24000:PRINT #2,"(U)p/(D)own
?";
15040 b$=INKEY$:b$=UPPER$(b$)
15050 IF b$="U" THEN CLS #2:GOTO 15120
15060 IF b$="D" THEN CLS #2:GOTO 15090
15070 GOTO 15040
15080 REM * * * down * * *
15090 FOR m=1 TO 7:GOSUB 15130:NEXT m
15100 GOTO 15210
15110 REM * * * up * * *
15120 GOSUB 15130:GOTO 15210
15130 FOR n=1 TO 8
15140 f(n)=d(char,n)
15150 NEXT n
15170 d(char,8)=f(1)
15180 FOR n=2 TO 8:d(char,n-1)=f(n):NEXT
 n
15190 RETURN
15200 REM * * * final * * *
15210 GOSUB 14560
```

```
15230 RETURN
15980 REM >>>>>>>> hori. shift <<<<<<<
16000 GOSUB 24000
16040 PRINT #2,"(L)eft/(R)ight";
16050 b$=INKEY$:b$=UPPER$(b$)
16060 IF b$="L" THEN CLS #2:GOTO 16100
16070 IF b$="R" THEN CLS #2:GOTO 16510
16080 GOTO 16050
16090 REM * * * left * * *
16100 FOR m=1 TO 7:GOSUB 16600:NEXT m
16110 GOSUB 14560
16120 RETURN
16500 REM * * * right * * *
16510 GOSUB 16600:GOSUB 14560
16520 RETURN
16590 REM * * * divide * * *
16600 FOR n=1 TO 8
16610 f(n)=d(char,n)
16620 x=f(n):x=INT(x/2):IF x<>f(n)/2 THE
N x=x+128
16630 d(char,n)=x
16640 NEXT n
16650 RETURN
16980 REM >>>>>>>>> exchange <<<<<<<<<
17000 GOSUB 24000
17010 char1=char
17050 GOSUB 24000
17070 FOR n=1 TO 8
17080 f(n)=d(char,n)
17090 NEXT n
17100 FOR n=1 TO 8
17110 d(char,n)=d(char1,n):d(char1,n)=f(
n)
17120 NEXT n
17130 GOSUB 14560
17140 RETURN
17980 REM >>>>>>>>>>> save <<<<<<<<<<<
18000 PRINT #2,"Filename ?"
18010 INPUT #2,n$
18020 n$="!"+n$
18030 SAVE n$,b,43776,256
18040 CLS #2
18050 RETURN
18980 REM >>>>>>>>>>> load <<<<<<<<<<<
19000 x=128:GOSUB 26000
```

```
19010 PRINT #2,"Filename ?"
19020 INPUT #2,n$:n$="!"+n$
19030 LOAD n$
19040 CLS #2
19050 GOSUB 21060:GOSUB 27000
19060 x=128:GOSUB 26000
19070 GOSUB 10000:GOSUB 22000
19080 RETURN
19480 REM >>>>>>>>> build pic <<<<<<<<
19500 GOSUB 19850
19540 GOSUB 24000:ch=char+223
19550 PRINT CHR$(23);CHR$(1);
19600 x=398:y=382
19610 MOVE x,y:PRINT #1,CHR$(ch);
19620 GOSUB 29000
19630 IF fi=1 THEN MOVE x,y:PRINT CHR$(2
3);CHR$(0);:PRINT #1," ";:PRINT CHR$(23)
;CHR$(1);
19635 MOVE x,y:PRINT #1,CHR$(ch);
19640 IF q=1 THEN GOTO 19700
19650 IF ex=1 THEN GOSUB 24000:ch=char+2
23
19660 x=x+16*(le=1 AND x>382)-16*(ri=1 A
ND x<590)
19670 y=y-16*(up=1 AND y<398)+16*(do=1 A
ND y>254)
19680 GOTO 19610
19700 GOSUB 19850
19710 GOSUB 10000
19720 RETURN
19840 REM * * * cls * * *
19850 PRINT CHR$(23);CHR$(0);:PAPER 0
19860 FOR n=1 TO 10
19870 LOCATE 24,n
19880 PRINT SPC(15)
19890 NEXT n
19900 PRINT CHR$(23);CHR$(1);
19910 RETURN
19980 REM >>>>> initialize screen <<<<<
20000 INK 0,0:INK 1,24:INK 2,3:INK 3,18
20010 PAPER 0:MODE 1:BORDER 0
20020 PAPER 2:FOR n=1 TO 25
20030 PRINT SPC(23):PRINT:NEXT n
20035 TAG #1
20040 PRINT CHR$(23);CHR$(1);
```

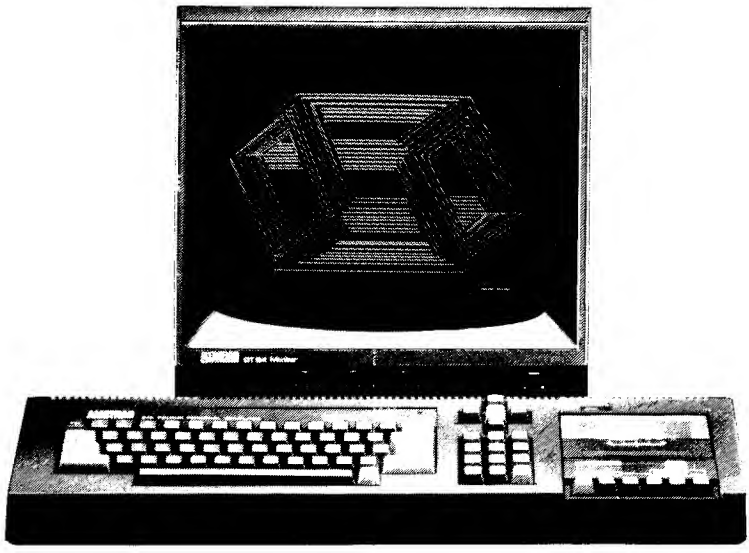
```
20050 PLOT 0,398,3
20060 PRINT #1,"No. Old New No. Old New"
;
20070 MOVE 0,382
20080 PRINT #1,"------------------------"
;
20110 y=358
20120 FOR n=1 TO 16
20130 MOVE 0,y:PRINT #1,223+n;CHR$(223+n
);
20140 MOVE 190,y:PRINT #1,239+n;CHR$(239
+n);
20150 y=y-22
20160 NEXT n
20170 GOSUB 27000
20180 x=128:GOSUB 26000
20190 GOSUB 25000
20210 WINDOW #2,25,39,11,13
20220 PAPER #2,1:PEN #2,0:CLS #2
20240 RESTORE 20310:y=180
20250 FOR n=1 TO 11
20260 READ n$:PLOT 382,y,3:PRINT #1,LEFT
$(n$,1);
20270 PLOT 398,y,1:PRINT #1,MID$(n$,2,LE
N(n$));
20280 y=y-16
20290 NEXT n
20300 PRINT CHR$(23);CHR$(1);:RETURN
20310 DATA Clear,Pick-up,Edit,Numbers,Mi
rror,Vert shift
20320 DATA Hori shift,X-change,Build,Sav
e,Load
20980 REM >>>> initialize variables <<<
21000 DIM f(8)
21010 DIM d(32,8)
21050 SYMBOL AFTER 224
21060 addr=43776
21070 FOR n=1 TO 32
21080 FOR m=1 TO 8
21090 d(n,m)=PEEK (addr)
21100 addr=addr+1
21110 NEXT m
21120 NEXT n
21130 addr=34999:GOSUB 29500
21250 RETURN
```

```
21980 REM >>>>>>>>>> draw char <<<<<<<<
22000 PRINT CHR$(23);CHR$(1);
22040 y=382
22050 FOR n=1 TO 8
22070 x=d(char,n):GOSUB 23000
22080 PLOT 382,y,3:PRINT #1,x$;x;
22090 y=y-16
22100 NEXT n
22110 RETURN
22980 REM >>>>>>>>>> X to X$ <<<<<<<<<<
23000 x$=BIN$(x,8)
23040 FOR m=1 TO 8
23050 IF MID$(x$,m,1)="0" THEN MID$(x$,m
,1)=" "
23060 IF MID$(x$,m,1)="1" THEN MID$(x$,m
,1)=CHR$(143)
23070 NEXT m
23080 RETURN
23980 REM >>>>>>>>>> input char <<<<<<<
24000 PRINT #2,"character ? ":PRINT #2
24010 n$=""
24040 FOR n=1 TO 3
24050 b$=INKEY$:IF b$<"0" OR b$>"9" THEN
 GOTO 24050
24060 PRINT #2,b$;:n$=n$+b$
24070 NEXT n
24075 CLS #2
24080 char=VAL(n$)
24085 IF char <224 OR char>255 THEN GOTO
 24000
24090 char=char-223
24100 RETURN
24980 REM >>>>>>>>>> draw grid <<<<<<<<<
25000 PRINT CHR$(23);CHR$(3);:x=382:y=38
2
25010 FOR n=1 TO 9:PLOT x,y,1:DRAW x+127
,y
25020 y=y-16
25030 NEXT n
25040 y=382
25050 FOR n=1 TO 9:PLOT x,y,1:DRAW x,y-1
25
25060 x=x+16
25070 NEXT n
25080 RETURN
```

```
25980 REM >>>>>>>> characters <<<<<<<<<
26000 PRINT CHR$(23);CHR$(1);:y=358:PLOT
 0,0,3
26040 FOR n=1 TO 16
26050 MOVE x,y
26060 PRINT #1," ";CHR$(223+n);:MOVE x+1
80,y:PRINT #1," ";CHR$(239+n);
26070 y=y-22
26080 NEXT n
26090 RETURN
26980 REM >>>>>>> array to chars <<<<<<
27000 addr=43776
27040 FOR n=1 TO 32:FOR m=1 TO 8
27050 POKE addr,d(n,m)
27060 addr=addr+1
27070 NEXT m:NEXT n
27080 RETURN
27980 REM >>>>>>>>>> X$ to X <<<<<<<<<<
28000 RESTORE 28090:x=0
28040 FOR m=1 TO 8
28050 READ y:w$=MID$(x$,m,1)
28060 IF w$<>" "THEN x=x+y
28070 NEXT m
28080 RETURN
28090 DATA 128,64,32,16,8,4,2,1
```

Now 'MERGE' the 'INKEYS' routine. . .

26
Picture Builder



Your Phosphor Palette

This is hi-tech art on your Amstrad, though it is unlikely that any of your 'masterpieces' will ever fetch thousands of pounds.

The program starts by asking for the mode (Ø, 1 or 2) and colours to be used for the border (Ø-26), paper (Ø 15) and the initial ink (Ø-15). A list of colour numbers is in your reference manual or in Appendix 5 of this book. A set of drawing colours are contained in the program. (Change the data in lines 2Ø11Ø and 2Ø12Ø for different colours.) The next request is for a picture name, which will be used for saving.

The final option offered is to have an 'information window' displayed. This will constantly show the current x,y co-ordinates of the drawing pointer and may be used to plan out plotting sequences to be incorporated into programs. The map of Great Britain in this book was generated in this way. The window appears at the top or bottom of the screen and automatically moves if it is in the way, whilst the picture underneath is still retained.

When the screen appears a graphic 'pencil' shows where the line

will be drawn. The pencil is moved using the joystick or cursor pad. Pressing the fire button or copy key changes the pencil to an eraser symbol. This will then rub out any lines that it passes over. If the fire button is pressed again a cross symbol will be displayed. This indicates that the cursor may be moved without drawing a line. Once in position to draw again, pressing the fire button will return to the pencil and draw mode. The cursor can be cycled through these modes by repeated pressing of the fire button.

A very useful alternative to using the pencil is to use the 'Auto line' mode. First the 'Move' symbol is obtained and moved to where the line is to start. Pressing 'Space' gives a beep and stores that position. The cross is then moved to the second location, 'Space' pressed and a line will be drawn between the two points.

At any time the current drawing ink can be changed using the 'Enter' key. Each press of the key steps through the available inks and the cursor changes colour to show the one selected.

'Shift' 'S' saves the current picture immediately to tape when selected. 'Record' and 'Play' should be pressed beforehand as prompts are not displayed.

'Shift' 'L' then 'Play' on the tape deck will load the first picture file found. The current screen will be cleared and the new one displayed.

Although drawing a picture can take some time, dependent upon the mode, producing 'art' in this way can be very enjoyable.

```
1 REM <<<< picture builder <> ISSI >>>>
2 REM
500 SYMBOL 240,240,200,164,146,79,47,30,
8
510 SYMBOL 241,240,226,228,152,24,36,66,
1
520 SYMBOL 242,240,184,220,238,119,56,27
,10
600 MEMORY 23699
610 SPEED WRITE 1
1000 GOSUB 21000
1010 x=0:y=399:dr=1:li=0
1020 hd=4:vd=2:TAG #3
1030 IF m=1 THEN hd=2
1040 IF m=2 THEN hd=1
1100 GOSUB 28000
1140 IF y<50 THEN po=395 ELSE po=20
1150 w$="x "+STR$(x)+"y "+STR$(y)+"ink "
+STR$(pe)
```

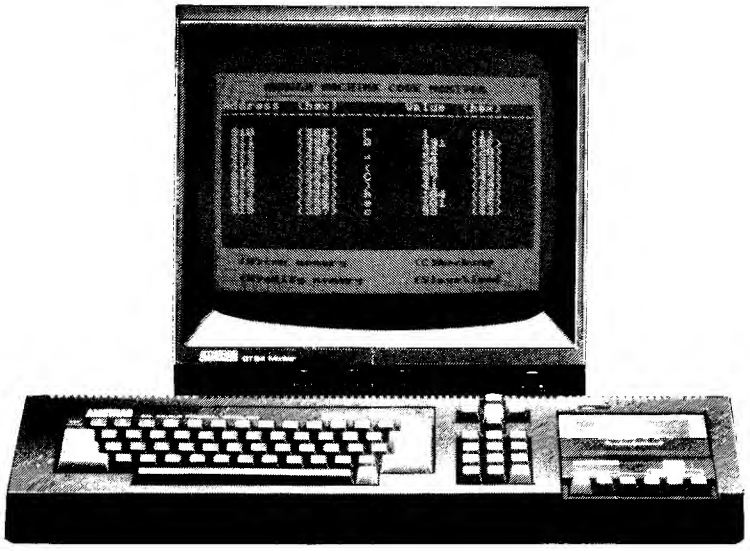
```
1155 PRINT CHR$(23);CHR$(1);
1160 IF inf=1 THEN MOVE 0,po:PRINT #3,w$
;
1200 PRINT CHR$(23);CHR$(1);
1210 PLOT x,y,pe
1220 PRINT #2,CHR$(240+dr);
1230 PRINT CHR$(23);CHR$(dr);
1240 IF dr=2 THEN ko=pap ELSE ko=pe
1250 PLOT x,y,pe
1260 PLOT x,y,ko
1370 PRINT CHR$(23);CHR$(1);
1380 PLOT x,y,pe
1390 PRINT #2,CHR$(240+dr);
1400 PRINT CHR$(23);CHR$(dr);
1440 PRINT CHR$(23);CHR$(1);
1450 IF inf=1 THEN MOVE 0,po:PRINT #3,w$
;
1460 IF sav=1 THEN GOSUB 18000
1470 IF lo=1 THEN GOSUB 18500
1480 IF li=1 THEN GOSUB 23000
1500 GOTO 1100
17980 REM >>>>>>>>>>> save <<<<<<<<<<<
18000 CALL 23700
18010 SAVE n$,b,&6AE0,&3FFF
18020 RETURN
18480 REM >>>>>>>>>>> load <<<<<<<<<<<
18500 LOAD "!"
18510 CALL 23715
18520 RETURN
19980 REM >>>>>> initialisation <<<<<<<
20000 MODE 1:PAPER 0:PEN 1
20010 INPUT "MODE ?",m
20020 IF m=0 THEN co=15:GOTO 20060
20030 IF m=1 THEN co=3 :GOTO 20060
20040 IF m=2 THEN co=1 :GOTO 20060
20050 GOTO 20010
20060 RESTORE 20110
20070 FOR n=0 TO co
20080 READ a
20090 INK n,a
20100 NEXT n
20110 DATA 0,18,24,2,6,20,15,3
20120 DATA 2,21,26,8,7,13,16,5
20130 CLS
20140 INPUT "BORDER ";bo
```

```
20150 IF bo<0 OR bo>27 THEN GOTO 20130
20160 BORDER bo
20170 CLS
20180 INPUT "PAPER ";pap
20190 IF pap<0 OR pap>27 THEN GOTO 20170
20200 PAPER pap
20210 CLS
20220 INPUT "INITIAL INK ";pe
20230 IF pe<0 OR pe>co THEN GOTO 20210
20240 PEN pe
20250 CLS
20260 INPUT "Name to 'SAVE' ",n$
20270 n$="!"+n$
20280 CLS
20290 INPUT "Information window ?",a$
20300 inf=0
20310 IF a$="y" OR a$="Y" THEN inf=1
20340 MODE m
20350 RETURN
20980 REM >>>>>>>>>> start <<<<<<<<<<<
21000 GOSUB 30000
21010 addr=23729:GOSUB 29500
21020 GOSUB 20000
21030 TAG #2
21100 RETURN
23000 SOUND 1,200,10,15
23010 IF flag=1 THEN GOTO 23050
23020 flag=1:x1=x:y1=y:dr=1
23030 td=hd:hd=hd*3:vd=6
23040 RETURN
23050 flag=0:hd=td:vd=2
23060 PRINT CHR$(23);CHR$(0);
23070 PLOT x,y,pe:DRAW x1,y1
23080 PRINT CHR$(23);CHR$(1);
23090 RETURN
27980 REM >>>>>>>>>> movement <<<<<<<<<
28000 GOSUB 29000:li=0:sav=0:lo=0
28010 IF le=1 AND x>0 THEN x=x-hd
28020 IF ri=1 AND x<639 THEN x=x+hd
28030 IF up=1 AND y<399 THEN y=y+vd
28040 IF do=1 AND y>0 THEN y=y-vd
28050 IF fi=1 THEN dr=dr+1:IF dr=3 THEN
dr=0
28060 PRINT CHR$(23);CHR$(dr);
28070 IF q=1 THEN pe=pe+1:IF pe=16 THEN
```

```
pe=0
28080 IF ex=1 THEN li=1
28090 IF a=67 THEN CLS
28100 IF a=83 THEN sav=1
28110 IF a=76 THEN lo=1
28120 RETURN
29980 REM >>>>>>>> poke b/move <<<<<<<<
30000 RESTORE 30060
30010 FOR n=23700 TO 23726
30020 READ a
30030 POKE n,a
30040 NEXT n
30050 RETURN
30060 DATA 33,0,192,17,224,106
30070 DATA 1,255,63,237,176,201,0,0,0
30080 DATA 33,224,106,17,0,192
30090 DATA 1,255,63,237,176,201
```

Now 'MERGE' the 'INKEYS' routine . . .

27
Arnold Monitor



Open RAM Surgery

This utility, which bears the Amstrad's original name, will help with investigations into memory. The machine's ROM routines can be looked at or any program which can co-reside in RAM can be examined. Memory can be modified, saved or loaded.

Command summary

View Memory. Asks for the start and finish locations, which have to be decimal. The memory contents will then be displayed in blocks of ten. 'Space' steps through to the 'finish'. 'Text' can be read from the ASCII columns.

Checksum. Asks for the start and finish. All the values are then totalled in decimal, inclusive of the addresses entered. This option is useful when entering from a listing which provides a 'Checksum' to check if errors have occurred.

Modify mem. Again the start and the end addresses are required. Then the display shows the current value. Pressing 'Enter' will retain

this value and the next location will be displayed. Typing numbers followed by 'Enter' will put what is typed into the current address. If the value is invalid then this will be indicated and no changes will be made to memory. The next location will then be shown. *Great care should be taken*, since *any RAM* may be modified with perhaps surprising results. (All entries are in decimal.)

Save. Asks for start and finish (decimal), then the file name. Saving commences when 'Enter' is pressed. 'Record' and 'Play' should be pressed prior to selecting the 'Save' option.

Load. This, when followed by pressing 'play', loads the first file found.

'Shift' 'Q' will quit any option.

```
1 REM <<<<<<< monitor <> ISSI >>>>>>>
2 REM
500 MEMORY 8500
510 SPEED WRITE 1
1000 GOSUB 20000
1500 PRINT #1,"Type Your Choice . . ."
1510 a$=INKEY$:a$=UPPER$(a$)
1520 IF a$="V" THEN GOSUB 21000
1530 IF a$="M" THEN GOSUB 22000
1540 IF a$="C" THEN GOSUB 23000
1550 IF a$="S" THEN GOSUB 24000
1560 GOTO 1510
19980 REM >>>>>>>>>> screen <<<<<<<<<<
20000 INK 0,0:INK 1,26:INK 2,18:INK 3,2
20010 MODE 1
20020 WINDOW #1,2,39,4,20
20030 PAPER 3:PEN 2:BORDER 0:CLS
20040 PAPER #1,0:PEN #1,1:CLS #1
20050 PRINT "       ARNOLD MACHINE CODE M
ONITOR"
20060 PRINT "      ----------------------
-------"
20070 LOCATE 4,22:PRINT "(V)iew memory"
20080 LOCATE 4,24:PRINT "(M)odify memory
"
20090 LOCATE 25,22:PRINT "(C)hecksum"
20100 LOCATE 25,24:PRINT "(S)ave\load"
20200 RETURN
20980 REM >>>>>>>>>>> view <<<<<<<<<<<
21000 GOSUB 26000
21070 CLS #1
```

```
21080 PRINT #1,"Address  (hex)          Va
lue  (hex)    ---------------------------
------------"
21090 FOR n=4 TO 13
21100 LOCATE #1,1,n:PRINT #1,sta
21110 LOCATE #1,10,n:PRINT #1,"(";HEX$(s
ta);")"
21120 LOCATE #1,24,n:PRINT #1,PEEK(sta)
21130 LOCATE #1,31,n:PRINT #1,"(";HEX$(P
EEK(sta));")"
21140 LOCATE #1,18,n:PRINT #1,CHR$(1);CH
R$(PEEK(sta));
21145 IF a$="Q" THEN GOTO 22230
21150 sta=sta+1:IF sta=fin+1 THEN GOTO 2
1200
21160 NEXT n
21170 a$=INKEY$
21175 IF a$="Q" THEN GOTO 21220
21180 IF a$<>" " THEN GOTO 21170
21190 GOTO 21070
21200 a$=INKEY$
21210 IF a$<>" " THEN GOTO 21200
21220 CLS #1:RETURN
21980 REM >>>>>>>>>> modify <<<<<<<<<<
22000 GOSUB 26000
22070 CLS #1
22080 PRINT #1,"Address  (hex)          Va
lue             ---------------------------
------------"
22090 FOR n=4 TO 13
22100 LOCATE #1,1,n:PRINT #1,sta
22110 LOCATE #1,10,n:PRINT #1,"(";HEX$(s
ta);")"
22120 LOCATE #1,24,n:PRINT #1,PEEK(sta)
22130 LOCATE #1,31,n
22140 a$=INKEY$:IF a$="" THEN GOTO 22140
22145 IF a$="Q" THEN GOTO 22230
22150 IF a$<"0" OR a$>"9" THEN GOTO 2216
0
22155 GOSUB 22500
22160 sta=sta+1:IF sta=fin+1 THEN 22210
22170 NEXT n
22180 a$=INKEY$
22190 IF a$<>" " THEN GOTO 22180
22200 GOTO 22070
```

```
22210 a$=INKEY$
22220 IF a$<>" " THEN GOTO 22210
22230 CLS #1:RETURN
22490 REM * * * mod * * *
22500 n$="":PRINT #1,a$;:n$=n$+a$
22510 GOSUB 22600
22520 IF a$=CHR$(13) THEN GOTO 22700
22530 GOTO 22510
22600 a$=INKEY$
22610 IF a$="" THEN GOTO 22600
22620 IF (a$<"0" OR a$>"9") AND a$<>CHR$
(13) THEN GOTO 22600
22630 IF a$=CHR$(13) THEN RETURN
22640 n$=n$+a$:PRINT #1,a$;
22650 RETURN
22700 a=VAL(n$):IF a<0 OR a>255 THEN SOU
ND 2,300,25,15:RETURN
22710 POKE (sta),a
22720 LOCATE #1,24,n:PRINT #1,a;"      "
22730 RETURN
22980 REM >>>>>>>>> checksum <<<<<<<<<
23000 GOSUB 26000
23010 tot=0
23020 FOR n=sta TO fin
23030 LOCATE #1,10,10
23040 PRINT #1,n
23050 tot=tot+PEEK(n)
23055 a$=INKEY$:IF a$="Q" THEN n=fin
23060 NEXT n
23070 SOUND 1,300,25,15
23080 SOUND 1,200,25,15
23090 SOUND 1,100,25,15
23120 LOCATE #1,10,12
23130 PRINT #1,"Check=";tot;
23140 a$=INKEY$
23150 IF a$<>" " THEN GOTO 23140
23160 CLS #1
23170 RETURN
23980 REM >>>>>>>>> save/load <<<<<<<<<
24000 PRINT #1,"  (S)ave or (L)oad ?
24010 a$=INKEY$:a$=UPPER$(a$)
24020 IF a$="S" THEN GOTO 24050
24030 IF a$="L" THEN GOTO 24500
24040 GOTO 24010
24050 CLS #1
```

```
24070 GOSUB 26000
24080 IF sta<8000 THEN GOTO 24050
24090 CLS #1:INPUT #1,"Name ";n$
24100 n$="!"+n$
24110 SAVE n$,b,sta,fin-sta+1
24120 CLS #1
24130 RETURN
24500 CLS #1
24510 INPUT #1,"Filename ";n$
24520 n$="!"+n$
24530 LOAD n$
24540 CLS #1
24550 RETURN
25980 REM >>>>>>>>>> input <<<<<<<<<<
26000 CLS #1
26010 PRINT #1," Start :"
26020 GOSUB 30000
26030 IF addr<0 OR addr>65535 THEN 26000
26040 sta=addr
26050 SOUND 2,300,25,15
26060 CLS #1:PRINT #1," Finish :"
26070 GOSUB 30000
26080 IF addr<0 OR addr>65535 THEN 26060
26090 fin=addr
26100 CLS #1
26110 IF sta>=fin THEN GOTO 26000
26150 RETURN
29980 REM >>>>>>>> input addr <<<<<<<<<
30000 n$=""
30010 a$=INKEY$
30020 IF (a$<"0" OR a$>"9") AND a$<>CHR$
(13) AND a$<>CHR$(127) THEN GOTO 30010
30025 IF a$=CHR$(127) AND LEN(n$)<1 THEN
 GOTO 30010
30030 IF a$=CHR$(127) THEN n$=LEFT$(n$,L
EN(n$)-1):GOTO 30050
30040 n$=n$+a$
30050 SOUND 2,150,5,15:LOCATE #1,10,10
30060 PRINT #1,n$;" "
30070 IF a$=CHR$(13) THEN GOTO 30090
30080 GOTO 30010
30090 addr=VAL(n$)
30100 RETURN
```

Final Words

In producing this book an attempt has been made to adopt a standard of game which is well above the norm.

This naturally leads to longer programs which require greater care to enter. The time will come therefore when you will need to debug your programs. This is when you have the opportunity to try to understand how and why the programs work.

To help you in this task there follows a 'debug guide'.

General points

1. Get someone else to double check that the entry matches the listing. Say it out loud.
2. Remember that in 40-column mode the listing should match the screen line by line.
3. Be careful that an I has not been confused with a 1.
4. Ensure that lower case 'a' is entered as such and not 'A'.
5. Watch out for ';' instead of ':'.
6. Double check spaces by reference to the spaces above and below.
7. Double check line numbers. Don't confuse a number which flows over onto another line as a line number.
8. Treble check data. It is very easy to skip a few numbers. As an overall check, count how many commas and compare with the listing.
9. Make sure that all quotes are in.
10. Is there really a fault? Re-read the instructions for play. Most responses need to be followed by the 'Enter' key.

Specific problems

1. To save space whilst still including substantial programs, extensive use is made of merging. Familiarise yourself with this technique on a simple program first. We anticipate some problems from readers in using this technique, but consider the finished results to be worth while for the majority.

2. One of the main sources of program failure is poor handling of bad entries. For example, a program may crash if an unexpected minus number were entered. It is possible to anticipate most 'silly' entries but only at the expense of needlessly longer listings. In writing these programs it has been assumed that the player will not set out to crash the program! They are therefore only protected against class three idiots. If you will be exposing your finished efforts to a hostile environment then it is up to you to protect against class one idiots.

3. All programs should work equally well on the monochrome Amstrad, but change the colours (after saving the correct version) to suit your own taste. *Micro Mind* can use symbols instead of colours to make the game playable in monochrome.

4. It is always good practice to load into a machine which has just been powered up. This can avoid problems if the previous program left something changed deep within the machine.

5. Unfortunately the Amstrad 'Random' function has been found to be very predictable. Care has been taken to minimise this problem whenever possible but you may be able to remember certain sequences.

Debugging

If none of the foregoing resolve the problem then roll up your sleeves and note some of these tips.

1. If an error message indicates a fault is in a particular line, but it looks right when examined, remember that the fault could be in the line which passes information to that line. Check back related lines.

2. Errors such as 'Out of range', 'Improper argument', etc., may be traced as above. To help type 'PRINT A' or whatever variables are used in that line. This should indicate the problem.

3. Bits being missed out! The Amstrad's 'Tron' function will show the order in which lines are called, often highlighting a rogue 'GOSUB' or similar problem.

4. Work methodically. Consider each section in turn. Set 'Break points' using 'End' or dump out variables by adding a short print line in the middle of the faulty routine.

5. Temporarily disable sections of program by adding a 'REM' to the start of a line. See what the effect is.

6. Re-type the section responsible! During development of these programs some 'Out of memory' errors could only be mysteriously solved by re-entering a single 'perfect' line.

7. Use the 'On error', 'Erl' and 'Err' functions normally as described in the manual.

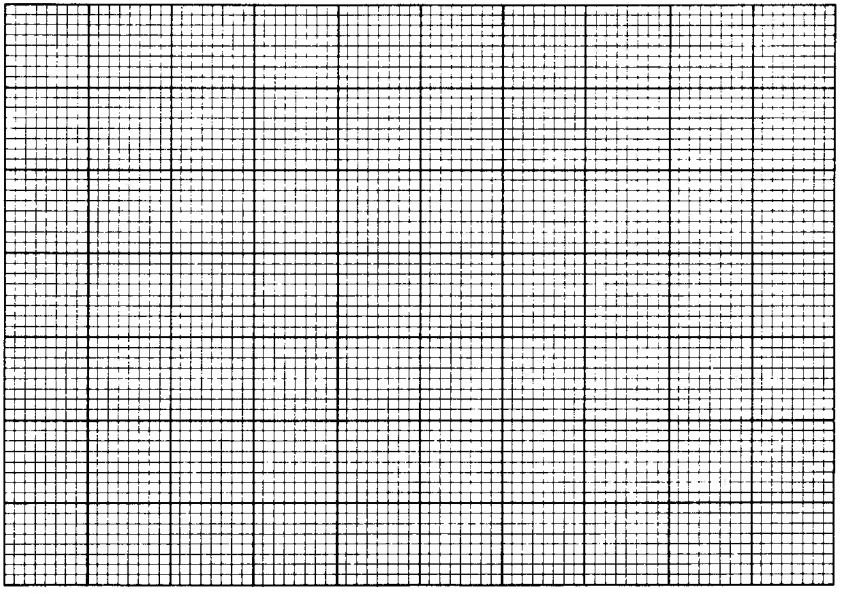
8. Go to bed – have a sleep. It sometimes works the next day!

Appendix 1: Complete Amstrad Character Codes

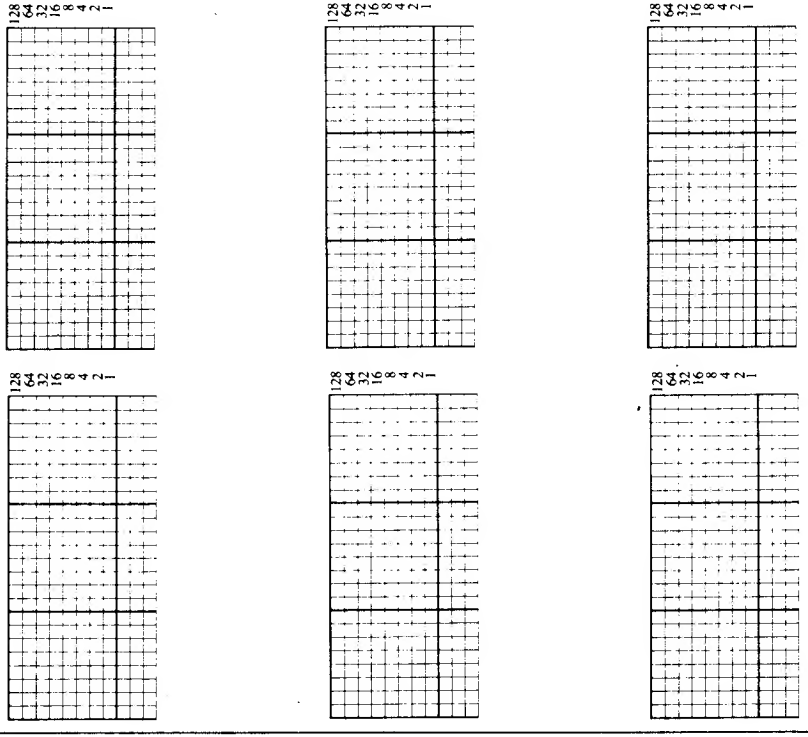
CHARACTER	CHR$	CHARACTER	CHR$	CHARACTER	CHR$	CHARACTER	CHR$
	0		32		64		96
	1		33		65		97
	2		34		66		98
	3		35		67		99
	4		36		68		100
	5		37		69		101
	6		38		70		102
	7		39		71		103
	8		40		72		104
	9		41		73		105
	10		42		74		106
	11		43		75		107
	12		44		76		108
	13		45		77		109
	14		46		78		110
	15		47		79		111
	16		48		80		112
	17		49		81		113
	18		50		82		114
	19		51		83		115
	20		52		84		116
	21		53		85		117
	22		54		86		118
	23		55		87		119
	24		56		88		120
	25		57		89		121
	26		58		90		122
	27		59		91		123
	28		60		92		124
	29		61		93		125
	30		62		94		126
	31		63		95		127

CHARACTER	CHR$	CHARACTER	CHR$	CHARACTER	CHR$	CHARACTER	CHR$
	128		160		192		224
	129		161		193		225
	130		162		194		226
	131		163		195		227
	132		164		196		228
	133		165		197		229
	134		166		198		230
	135		167		199		231
	136		168		200		232
	137		169		201		233
	138		170		202		234
	139		171		203		235
	140		172		204		236
	141		173		205		237
	142		174		206		238
	143		175		207		239
	144		176		208		240
	145		177		209		241
	146		178		210		242
	147		179		211		243
	148		180		212		244
	149		181		213		245
	150		182		214		246
	151		183		215		247
	152		184		216		248
	153		185		217		249
	154		186		218		250
	155		187		219		251
	156		188		220		252
	157		189		221		253
	158		190		222		254
	159		191		223		255

Appendix 2: Defined Character Design Aid

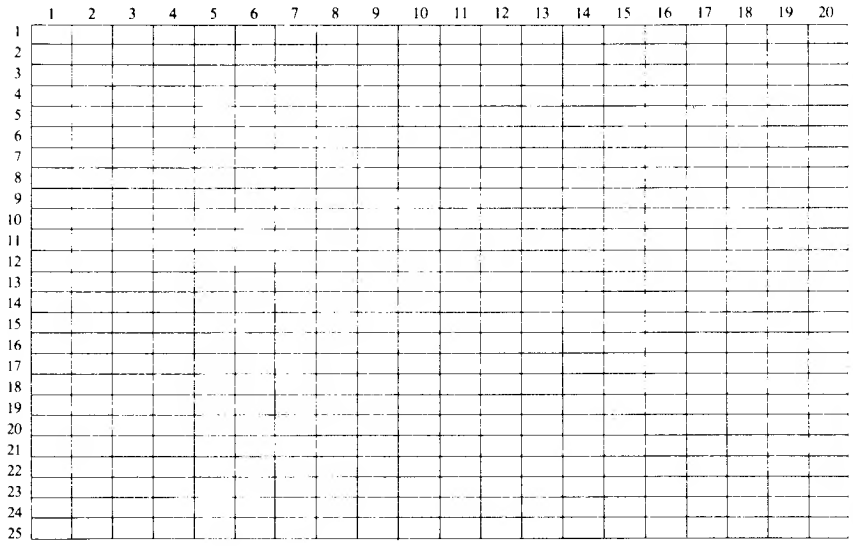


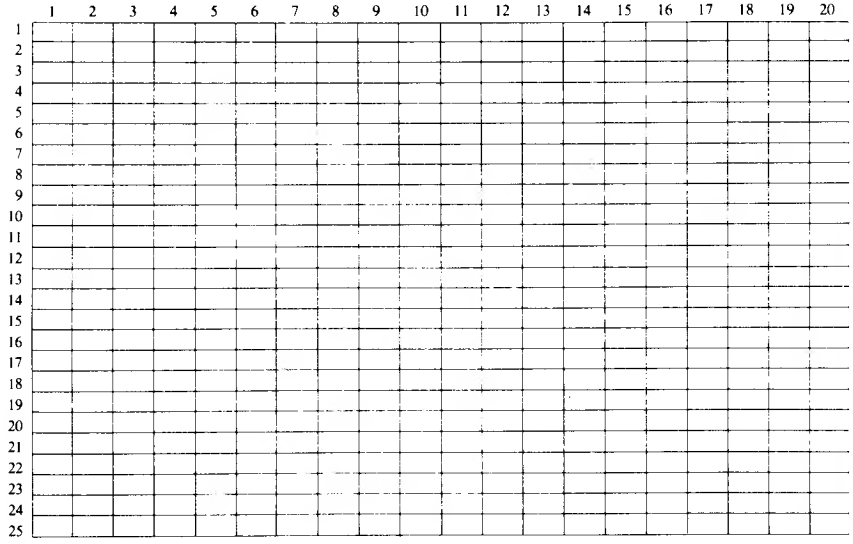
CHARACTER DEFINITIONS



Appendix 3: Screen Layout Planner

MODE Ø 20 COLUMNS x 25 ROWS 16 COLOURS

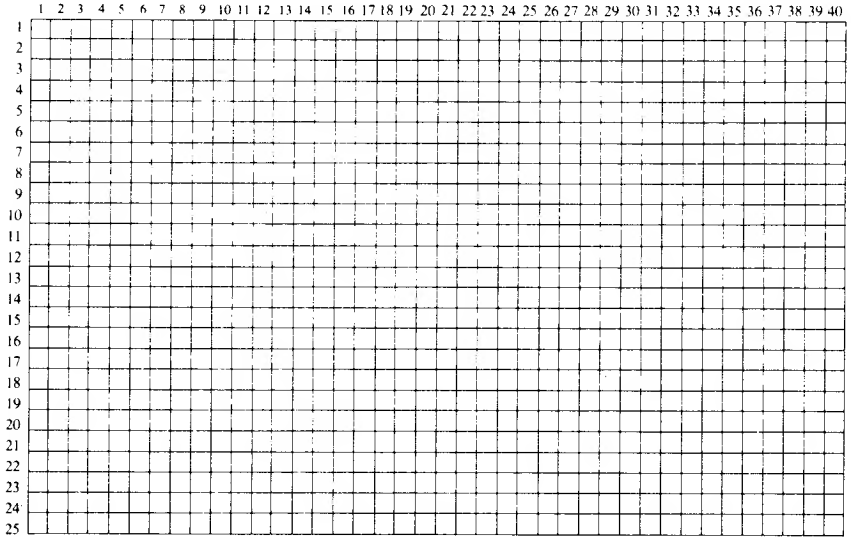


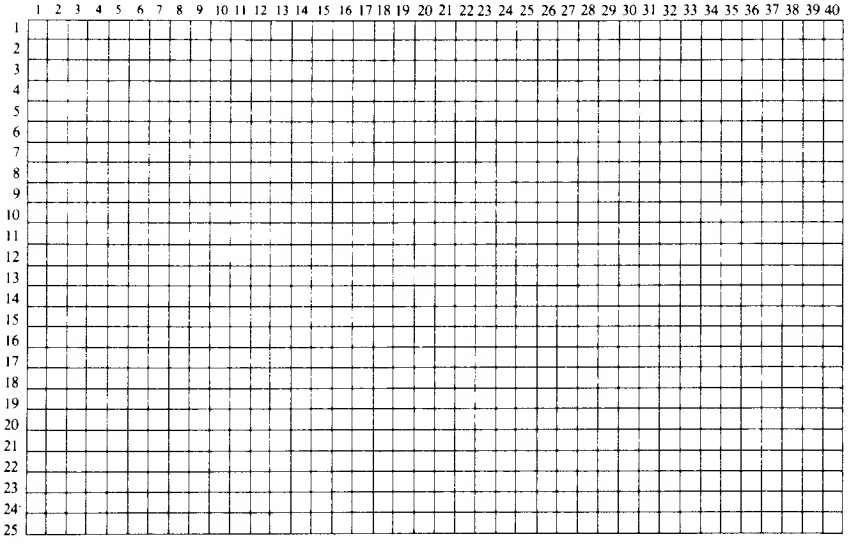


COLOUR ALLOCATION

PEN	Ø =	6 =	11 =
	1 =	7 =	12 =
	2 =	8 =	13 =
	3 =	9 =	14 =
	4 =	10 =	15 =
	5 =		

MODE 1 40 COLUMNS x 25 ROWS 4 COLOURS



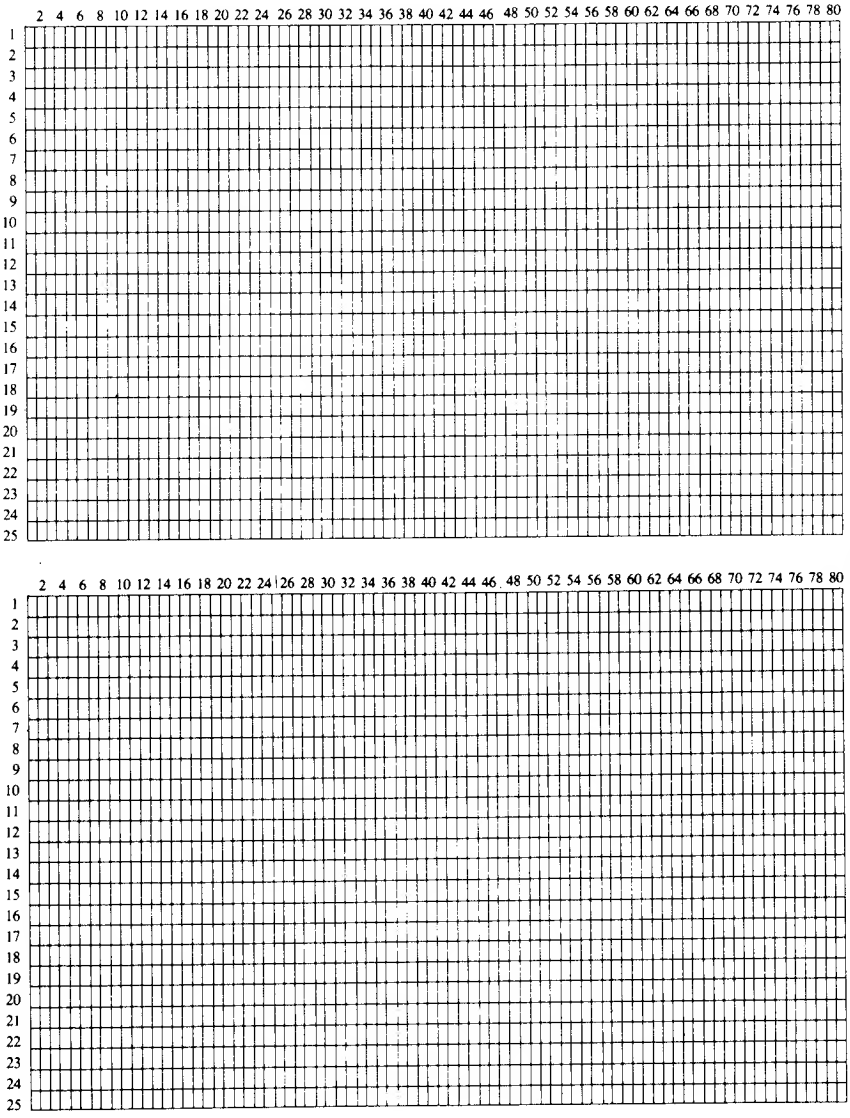


COLOUR ALLOCATION

PEN Ø = 2 =

1 = 3 =

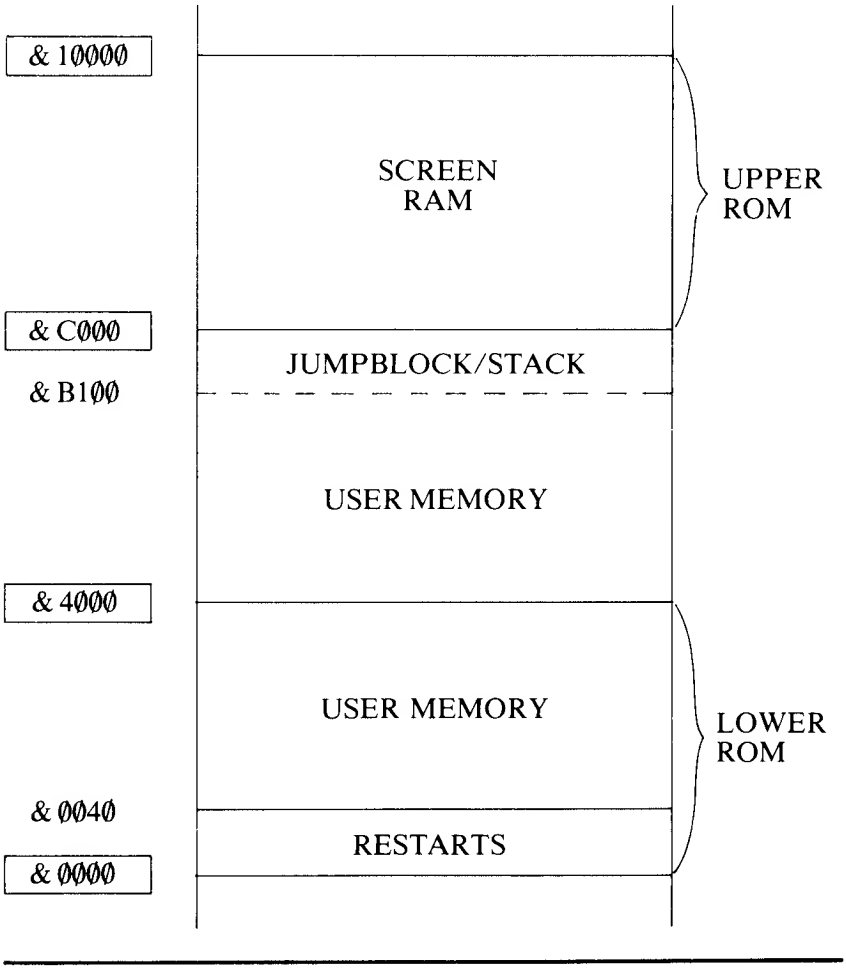
MODE 2 80 COLUMNS x 25 ROWS 2 COLOURS



COLOUR ALLOCATION

PEN 1 = PEN 2 =

Appendix 4: Amstrad Memory Map



Appendix 5: Amstrad Colour Table

	COLOUR		COLOUR
0	BLACK	13	WHITE
1	BLUE	14	PALE BLUE
2	BRIGHT BLUE	15	ORANGE
3	RED	16	PINK
4	MAGENTA	17	PALE MAGENTA
5	MAUVE	18	BRIGHT GREEN
6	BRIGHT RED	19	SEA GREEN
7	PURPLE	20	BRIGHT CYAN
8	BRIGHT MAGENTA	21	LIME GREEN
9	GREEN	22	PALE GREEN
10	CYAN	23	PALE CYAN
11	SKY BLUE	24	BRIGHT YELLOW
12	YELLOW	25	PALE YELLOW
		26	BRIGHT WHITE

Appendix 6: Quick Syntax Checker

The Amstrad manual contains a comprehensive and instructional section for each part of the BASIC syntax. However, sometimes it is useful to be able to have an idea of which 'word' is required before looking it up. This list of all the keywords has been separated into useful subsections. Anyone wishing to learn the commands associated with say 'SOUND' would simply look up the correct list and then refer to the individual section in the Amstrad manual.

MATHS
ABS
AND
ATN
CINT
COS
CREAL
DEF
DEG
EXP
FIX
FN
INT
LOG
LOG 10
MAX
IN
MOD
NOT
OR
PI
RAD
RANDOMIZE
RND
ROUND
SGN
SIN
SQR
TAN
UNT
XDR

GRAPHICS
CLG
DRAW
DRAWR
MOVE
MOVER
ORIGIN
PLOT
PLOTR
TAG
TAGOFF
TEST
TESTR

CONTROL
AFTER
CALL
DI
EI
ELSE
END
ERROR
EVERY
FOR
GOSUB
GOTO
IF
INKEY
INKEY$
IMP
JOY
LINE INPUT
NEXT
ON
ON BREAK
ON ERROR GOTO
OUT
PEEK
POKE
REMAIN
RETURN
SPEEDINK
SPEEDWRITE
STEP
STOP
THEN
TIME
TO
WAIT
WEND
WHILE
WIDTH

SOUND
ENT
ENV
ON SQ
RELEASE
SOUND
SQ

PROGRAMMING
AUTO
CLEAR
CONT
DATA
DELETE
DEFINT
DEFREAL
DEFSTR
DIM
EDIT
ERASE

ERL
ERR
FRE
HIMEM
INPUT
KEY
KEYDEF
LET
LIST
MEMORY
NEW
READ
REM
RENUM
RESTORE
RESUME
RUN
SYMBOL
SYMBOL AFTER
TROFF
TRON

FILES
CAT
CHAIN
CLOSEIN
CLOSEOUT
EOF
LOAD
MERGE
OPENIN
OPENOUT
SAVE

STRING
ASC
BIN$
CHR$
DEFSTR
HEX$
INSTR
LEFT$
LEW
LOWER$
MID$
RIGHT$
SPACE$
STR$
STRIN$
UPPER$
VAL

SCREEN
BORDER
CLS
INK
LOCATE
MODE
PAPER
PEN
POS
PRINT
SPC
TAB
USING
VPOS
WINDOW
WINDOW SWAP
XPOS
YPOS
ZONE

Here is a selection of 24 exciting, high-quality games written especially for the Amstrad and designed so the reader can create his own games using the basic subroutines in the book. All categories of games are included from the classic Maze Maniac to educational games such as Sum Fun. Each program is presented to appeal to all Amstrad owners no matter how old or young, whether you are completely new to computing or fairly experienced. A selection of useful routines is also given to help you modify the memory, create original graphics and develop your own programs. A reference section includes character codes, memory map, quick syntax checker and screen and char grid.

All the games make full use of the Amstrad's facilities and are thoroughly tested and crash-proofed.

The Author

Jim Gregory is Managing Director of Mr Micro, one of Britain's leading games software houses. He has contributed to *Personal Computer World, Leisure and Electronics Trader* and *Computer and Video Games.*

Front cover illustration by Jeff Ridge

COLLINS

Printed in Great Britain

0 00 383121 3

£5.95 net

ADDITIONAL NOTES FOR:

SENSATIONAL GAMES FOR THE AMSTRAD

Since publication of the above book some problems have been reported by purchasers. Each report has been carefully examined and these additional notes have been produced to help ease any difficulties and even improve some of the listings. If a program title does not appear in these notes then it may be assumed that they have not caused any problems.

MARIE CELESTE

There are no faults in the listing but there has been confusion over the symbol printed between quotes for V$. This is the apostrophe and is obtained using [shift] 7.
To remove the ability to cheat the following lines can be added.

```
34065 IF O(N)<>PO THEN 34040
36020 IF N$=O$(N) THEN 36055
36055 IF O(N)<>PO THEN 36040
```

WARNING ONLY TO BE READ BY THOSE WHO HAVE DIFFICULTY SOLVING THE ADVENTURE:
We suggest you start by taking the knife then climbing the rope. DO NOT GO AFT YET or you may be trapped. Once you have cleaned the decks and found a new location you should be able to solve the rest. NO FURTHER CLUES WILL BE GIVEN!

RAINBOW BREAKOUT

Listing is correct but problems have arisen by confusion of l's with I's so take care!

STAR TREK

Listing is correct except for the minor error in lines 27130 and 27100 in which CHR$(235) should be changed to CHR$(234). This will produce the correct symbol to shoot at!

DUO DRAUGHTS

Problems reported such as not clearing, extra draughts etc. are due to entry mistakes.
Note: Stalemate situation cannot be detected by the program and so 'QUIT' must be used to end such games.
The board should be printed as shades of blue for draughts and shades of red for Chess - this is not a fault.

There is a problem with the 'REPLAY' option which can be corrected by the addition of the following lines.

```
11065 Z$(count,1)="FF"
11080 IF A$="Y" THEN DEM=1:GOSUB 35005:GOTO 1100
11090 IF A$="N" THEN GOTO 1100
34160 GOTO 28100
35000 DEM=0
35005 COUNT=0:PLAY=1
35500 IF DEM=1 THEN RETURN
35505 IF PLAY=1 THEN 29000
```

PONTOON

There are some lines which disappeared during make up of the book. They should be added to the lines already listed.

```
11100 C=(RND(1)*51)+1:IF AV(C)=0 THEN 11100
11105 NC=NC+1:BC(NC)=C
11110 RETURN
```

On page 100 Line 16011 is not clear. It actually ends "..... THEN 16005"

On page 98 line 65 should end with just 80(the '66' should not be entered)

PICK MAN

This is correct as listed. Unfortunately extensive machine code would be required to speed up the game. None the less it is still a playable BASIC game.

SKIPPY

To prevent problems if the log carries SKIPPY off screen the following lines should be added:

```
463 IF KX+KDX=0 OR KX+KDX=20 THEN 50500
710 MODE 0:INK 15,24:INK 14,26:BORDER 0:PAPER 0:CLS:PRINT
CHR$(30):INK 0,0
```

TODAY ENGLAND

The program is correct as listed and all problems have been from incorrect DATA which gives a syntax error (or a strange map) when typed in wrongly.
If you do not want the response box to flash enter:

```
905 PAPER#3,0
```

If you want cities to be removed once guessed then replace the existing lines with these ammendments:

```
460 PLOT X,Y,3
843 PLOT X,Y,3
```

CHARACTER BUILDER

To check that characters are in range when requested by the operator, the following line should be added:

```
13090 IF VA<0 OR VA>255 THEN 13050
```

Disk Owners will have to alter the memory area used since it has been found to clash with the D.O.S. In lines 18030,21060 and 27000 the number 43776 should be changed to 42492.

Note: All other programs have now been tested with the Amstrad Disc System and work perfectly.

GENERAL PROBLEMS

IMPROPER ARGUMENTS

The vast majority of feedback obtained has been concerning the error message " IMPROPER ARGUMENT" This is when a recognised command cannot be executed. In particular this occurs on SYMBOL AFTER instructions. The reason being that if this is attempted twice (i.e. because the program has already run once) the ram top will already be set and so the error will result.

The solution is first of all not to run the program until completed and saved. If however you have mistyped something and the program breaks out, then 'DO NOT RUN' again after correcting the error.

Instead it is best to 'fool' the machine into re-setting by starting a 'LOAD' sequence. This may be most simply done by holding down 'CTRL' and pressing 'ENTER'.

This will produce the 'press play' prompt. Next use the escape key to break out. The program can then be run again.

Unfortunately the IMPROPER ARGUMENT is a feature of AMSTRAD'S operation which is not easy to overcome, other than described above.

SYNTAX ERRORS

When the machine gives a syntax error it does not mean that the listing is wrong. Follow the hints given in the book to resolve any problems. Take particular care entering DATA lines since this is far and away the most common cause of programs not functioning as expected.

I thank you for purchasing the book and hope that these additional notes will increase your enjoyment of it.

JIM GREGORY DEC. '84